UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

Continuous Security Assessment

Pedro Miguel da Costa Santos

Mestrado em Segurança Informática

Dissertação orientada por:
Prof. Doutora Maria Dulce Pedroso Domingos
e por Eng. José António Dos Santos Alegria

2018

Agradecimentos

A realização de uma dissertação de mestrado é como percorrer um trajeto sinuoso sem ajuda, e que apenas se torna mais difícil sem o apoio de outros. Esta seção é, portanto, dedicada a todos os que, de forma direta ou indireta, contribuíram para a realização deste projeto.

Não posso começar sem primeiro agradecer à professora Doutora Dulce Domingos, por toda a paciência que teve para comigo, por todo o conhecimento que me transmitiu e por todo o tempo que dedicou a este projeto, foi essencial para a concretização do mesmo.

Em segundo lugar, quero agradecer ao orientador da Portugal Telecom, o Engenheiro José Alegria, pela oportunidade de realizar o estágio no seio da segurança informática da PT. Quero ainda agradecer todo o tempo despendido da sua limitada agenda e toda a sabedoria partilhada.

Um agradecimento especial ao Jorge Silva, que foi sem dúvida essencial para a progressão desta tese na vertente técnica. Por todo o tempo prestado, pelas dicas, pelo seu incentivo, disponibilidade e apoio que sempre demonstrou, expresso-lhe aqui a minha gratidão.

Quero ainda agradecer ao Pedro Gonçalves e Pedro Inácio que, de forma direta, contribuíram para o meu ingresso no projeto. O meu sincero obrigado por todo o aconselhamento transmitido, contribuindo assim para o meu enriquecimento tanto a nível pessoal como profissional. Uma nota de agradecimento para a DCY em geral pelo ambiente familiar com que me acolheram desde o primeiro minuto.

Uma nota também de agradecimento ao João Santos pela amizade, e que sendo um ex-estagiário me guiou nesta aventura. Um agradecimento em especial para a Marisa Félix, colega de estágio, por toda a amizade, aconselhamento, e ainda porque tornou toda esta experiência mais fácil de lidar.

Um sentido obrigado à minha família, nomeadamente aos meus pais, ao meu irmão e cunhada, aos meus avós, aos meus sogros, à Náná e ao Belo, e todos os restantes, que me tornaram na pessoa que sou hoje, e que são indispensáveis para a pessoa quero continuar a ser. Tenho ainda a noção de que a realização deste mestrado foi duro para convosco, e nunca vos irei conseguir agradecer o suficiente, amo-vos a todos. Um sincero obrigado. Amo-te mãe! Amo-te pai!

Por fim, a pessoa a quem este mestrado mais prejudicou, porque nos momentos mais fáceis mas principalmente nos mais difíceis estiveste sempre a meu lado, este mestrado é também para ti. Não existem palavras para caracterizar o que sinto por ti. Quero que saibas que uma vida a teu lado é pouco, mas vamos usufruir dela ao máximo. Amo-te meu amor. Amo-te minha Patrícia.

O meu profundo agradecimento a todas as pessoas que contribuíram para a concretização desta dissertação, quer fosse a nível intelectual, quer fosse a nível emocionalmente.

À Patrícia, aos meus pais, à família.

Resumo

Nos dias de hoje, o tema da segurança da informação é um tema sensível para qualquer organização. Hoje em dia, é cada vez mais fácil para um indivíduo com intenções maliciosas obter acesso a informação ilegítima e/ ou causar danos aos dados disponibilizados pelas empresas. Numa altura em que os ataques informáticos são cada vez mais elaborados, é vital para qualquer entidade possuir uma defesa física e tecnológica robusta. A cada mês que passa, o número de vulnerabilidades descobertas facilmente supera o número do mês anterior e é ultrapassado pelo número do mês seguinte[36][60], e a falta de sensibilidade na matéria pode levar à exploração de uma vulnerabilidade com sucesso por parte de um individuo mal-intencionado. Qualquer organização deve estar ativamente em prevenção sobre as suas infraestruturas, por forma a precaver qualquer vulnerabilidade que possa vir a ser descoberta, e prevenindo que a mesma seja explorada, não apenas devido a obrigações legais, mas também pelo impacto que um ataque possa ter na imagem e negócio da mesma organização. Nasce desta forma uma nova área nas organizações com fim a proteger as mesmas desta nova realidade, a cibersegurança.

A cibersegurança consiste no conjunto de pessoas, processos e práticas que visa a proteção de redes, computadores, programas e informação de ataque, dano ou acesso ilícito[41]. Esta é avaliada principalmente através de três propriedades: confidencialidade, disponibilidade e integridade[14]. Qualquer uma destas três propriedades quando violada coloca todo o ecossistema da organização em risco.

Este projeto denominado de Continuous Security Assessment, encontra-se no âmbito da dissertação do Mestrado em Segurança Informática (MSI) da Faculdade de Ciências da Universidade de Lisboa (FCUL), e resulta de uma parceria entre a Fundação da Faculdade de Ciências da Universidade de Lisboa(FFCUL) e a Portugal Telecom (PT)[56]. A PT, sendo uma das principais entidades a operar em Portugal, possui uma direção dedicada à segurança e privacidade da informação, a Direção de Cyber Security and Privacy (DCY). A DCY detém vários projetos que visam a segurança da infraestrutura da PT, um destes é o CyberWatch.

O projeto CyberWatch tem o propósito de controlar e rever de forma contínua a cibersegurança do ecossistema PT, mais concretamente, abrange um conjunto de processos que visam consciencializar os recursos quanto a novas ameaças na web, novas vulnerabilidades, como ainda no aperfeiçoamento das pessoas, processos e tecnologias. Um dos

processos do projeto CyberWatch consiste na deteção, reporte e acompanhamento da evolução das vulnerabilidades nos ativos PT, processo em que esta tese está inserida, mais especificamente, visa a descoberta e acompanhamento de vulnerabilidades recorrendo ao uso de ferramentas de *vulnerability scanning*.

No ano de 2015/2016 a FCUL e a PT associaram-se e foi realizado o projeto "*Desenvolvimento de um processo automático de Gestão de Vulnerabilidades de Cibersegurança em ambientes de grande dimensão*", cujos objetivos foram os seguintes:

- Ponto central para gerir o agendamento de scans de vulnerabilidades a ativos PT.
- Gestão central dos vários scanners de vulnerabilidades à disposição da PT, independentemente da tecnologia.
- Transmissão dos resultados para os repositórios de dados da DCY.

O resultado final deste projeto foi a criação de uma plataforma cujo nome é Vulnerability Assessment Coordinator (VAC). O VAC foi desenvolvido tendo por base as necessidades da PT na altura. O projeto de Continuous Security Assessment, é uma extensão significativa do projeto descrito anteriormente, agora enquadrado no programa de desenvolvimento CyberWatch, e visa numa primeira instância responder às limitações/problemas atuais do VAC:

- Dificuldade no uso da ferramenta.
- Falta de interoperabilidade com outras tecnologias de *vulnerability scanning*.
- Controlo e integração com os repositórios de informação da PT.
- Impossibilidade de integração dos resultados num contexto do *software* Maltego.[1]

Tendo em consideração os problemas descritos anteriormente, esta tese propôs-se a atingir as seguintes contribuições:

- Melhoria do *software* VAC - esta transformação dará origem a uma nova versão do *software* que se passará a denominar por VACv2-, mais concretamente, tornar a ferramenta escalável quanto a tecnologias de *vulnerability scanning* e tipos de scan, melhorar a usabilidade da ferramenta, e proteger a própria ferramenta de acessos ilícitos.
- Adição de uma nova tecnologia de *vulnerability scanning* ao VACv2 - Qualys Cloud-based.

[1] O Maltego trata-se de um *software* especializado em correlação de dados.

- Melhoria do módulo de resultados do VACv2, e providenciar o controlo do mesmo ao utilizador. Também inserido neste ponto está o correlacionamento dos resultados dos *scans* por parte do Maltego.

O VACv2 é um esforço para evitar que todo o trabalho dispensado na realização do VAC seja em vão. O VACv2 permite um maior facilitismo na configuração de *scans* periódicos, não sendo agora imputada responsabilidade ao operador por fazer a gestão dos ativos por cada plataforma de *vulnerability scanning*, e ainda, no tratamento dos resultados dos *scans*, sendo que é também objetivo que o VACv2 permita uma gestão centralizada das vulnerabilidades existentes em toda a infraestrutura da PT. Por fim, algo de inovador é o Maltego, *software* que visa a correlação de dados, recolher os dados associados aos resultados *scans* e correlacionar os mesmos com os dados vindos de outras plataformas a fim de facilitar o trabalho de outras equipas que têm a função de responder e mitigar eventos de segurança na PT, nomeadamente a equipa do Security Operations Center(SOC).

Palavras-chave: Cibersegurança; CyberWatch; Vulnerabilidades; VACv2

Abstract

Nowadays, Information Security is a sensitive issue for any company. Portugal Telecom is one of the most influential companies in Portugal, and to provide confidence to the general public, partners, and stakeholders, it must have a well-defined procedure for securing information. One of PT's procedures to prevent potential attacks is to check its infrastructures for unknown vulnerabilities periodically. To help to find vulnerabilities present in their systems, PT has acquired scanning technologies. However, these technologies are highly dependent on human interaction.

In the past, there has been one thesis in which the final solution - which name is Vulnerability Assessment Coordinator (VAC) - consisted of a centralized point for managing and orchestrating scans of PT's critical assets. However, this software never made it to a production environment due to its growing limitations. It is in this context that this master thesis - Continuous Security Assessment - is located. This thesis proposes to achieve the following goals:

- Improvement of VAC Overall enhancement of the tool, making its scalable regarding technologies, improving usability, and protecting its data from unwanted access;
- Addition of a new Scanning Technology into VACv2 addition of the new scanning technology, Qualys, and meeting its requirements;
- Results treating & Data correlation - Improving the results manager module and providing control to the operator, and allowing data correlation.

This master thesis will take advantage of the software already developed by the other master thesis and will improve the usability of it, while also making it independent of any scanning technologies and scanning types. One other feature that will be intended for this thesis is the possibility of correlating the scan results provided by the scanning technologies with other sources with resort to specialized software for data correlation.

Keywords: Vulnerabilities; Continuous Security Assessment; VAC; Centralized; Scanning Technologies

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The Internet was an unprecedented achievement that until this day is still revolutionizing the world in new unimaginable ways. The last few years were crucial, an example of such is the introduction of the term "Internet-of-Things" which refers to the time of when more "things/devices" will be connected to the internet than humans. However, and this comes with no surprise, most of the devices available to the general public are vulnerable concerning cybersecurity. Let's take for example the waterfall model, which consists in dividing the development of a product into multiple stages. Most companies would consider security as being a step in this model. However, the "security" of the product should be enforced in every step of this model, not only because security aspects change according to the stage the product is, but also because security is not something that one adds into the system, like another functionality. Security must be present from the starting of the product until the end of its lifecycle.

Because companies can be both client and manufacturer, they need to be aware of vulnerabilities in their systems, not only for legal and criminal reasons but also because of the impact that a successful attack could have into the business and image of the organization. To help companies being aware of their vulnerabilities, some organizations are focused on offering solutions that its purpose is to find vulnerabilities in systems or websites.

Portugal Telecom (PT) being one of the most influential companies in Portugal, and a reference in the telecommunications area, to provide confidence to the general public, partners and stakeholders, it has a well-defined procedure for the discovery of vulnerabilities and an action plan for when one is encountered, to prevent the exploitation of such vulnerability by a malicious attacker. To find these vulnerabilities, PT has acquired a few technologies for discovering vulnerabilities. However, these technologies are highly dependent on human interaction.

It is in this context that this master thesis - Continuous Security Assessment - is lo-

cated, it is intended that the project of this thesis is based on one solution previously developed which was the project of another thesis, the software was named Vulnerability Assessment Coordinator, and is a centralized point for managing and orchestrating scans to PT's critical assets.

1.2 Objectives

The project of Continuous Security Assessment intends to improve the actual procedure of vulnerability assessment done by Portugal Telecom, this is assured by PT's Cybersecurity Direction (DCY). The procedure of vulnerability assessment is currently being done on critical assets only, and with resort to two scanning technologies, OpenVAS and Nexpose. However, the process associated with the management of actual scans, or configuration of new scans is exceptionally time-wasting becoming inefficient the use of the scanning technologies, not to mention that the procedures associated with this tasks are incredibly repetitive.

For those reasons, DCY has thought in developing a software in which it would manage the scanning technologies autonomously and orchestrate the launch of scans, while also uploading the results into DCY's information repositories systems. This software was the final product of one other master thesis, and the final name of the solution was Vulnerability Assessment Coordinator (VAC), but VAC got deprecated even before starting its operation for two reasons. The first was that it was built to work with the OpenVAS and Nexpose scanning technologies only, and the second is that it was incredibly difficult to use the full capabilities of the solution.

The Continuous Security Assessment project is the result of PT deciding not to waste all the effort spent in VAC, this project's objectives are:

Improvement of VAC Overall enhancement of the tool, making its scalable regarding technologies, improving usability, and protecting its data from unwanted access;

Addition of a new Scanning Technology into VACv2 addition of the new scanning technology, Qualys, and meeting its requirements;

Results treating & Data correlation - Improving the results manager module and providing control to the operator.

This project comes at a time in which PT is acquiring a new scanning technology and is letting go Nexpose, thence the disassociation of scanning technologies from VAC, to avoid the need of development if this situation occurs again in the future. Also, the project will seek to automate the procedure of vulnerability assessment currently handly by DCY's personnel something that the earlier version of VAC failed.

However, the last objective is not focused on the same technology as the remaining. The last objective will be over software that is focused on the correlation of information.

This software, named Maltego, will be handled mainly by the Security Operation Center team (SOC), what this last objective pretends it to allow the correlation of data from other sources of PT with the results produced by the scanning technologies, something that until this moment was not being done.

1.3 Contributions

This thesis actively contributed to the improvement of a tool - VAC - that was being unused by PT, and allowed a more natural data correlation between the results produced by such platform and other sources of information. Let's have a look at the contribution achieved by this thesis.

VAC changed its name into VACv2 because of the project done by this thesis. Concerning VAC, VACv2 is now more secure thanks to the ciphered communication client-server, authentication of users against PT's Active Directory through the Kerberos protocol, and checking if they have permission to access VACv2. The tool is now more user-friendly by allowing the user to edit any object created - something that VAC was not capable of -, and also fetches all the data possible in order to present to the user when configuring any object in the platform - something VAC did not do. VACv2 is now detached from any scanning technology or scanning type, it is also detached from any information repository belonging to PT, and the operator has control of every single object in the platform. This master thesis had one other feature which was a development not directly involved with the primary software - VACv2. This development consisted in extending the capabilities of a specialized software in correlating data - named Maltego -, by making it fetch the results of the scans provided by VACv2 and correlating it with other sources of information.

In PT's real environment the project will contribute to at least four entities, let's analyze them in more detail:

DCY/Cybersecurity Engineering Team (CSE) - This is the team, that will gain most with VACv2 development and deployment to a production environment, because it is this team's responsibility to perform the tasks that VACv2 will automize, tasks like the scan configuration, scan management across the different scanning technologies, and scan schedulings.

DCY/Vulnerability Assessment and Management Team (VAM) - This is the team which is currently doing all the post-processing of the scan results produced by the different scanning technologies, which if the results come from different platforms, it would be normal if they had different structures. VACv2 will not only uniformize these results into a single format but will also be a centralized point for managing the existing vulnerabilities. In other words, VACv2 will perform all the post-

processing of the results accordingly to what is configured in the system and will upload the values into DCY's data analytics platforms.

DCY/Computer Security Incident Response Team (CSIRT) - In a nutshell, VACv2 will upload its values into DCY's data analytics platforms. From these platforms, it will be possible for a specific software - Maltego - to fetch the scan results, to correlate the scans data with information provided by other sources, this way easing CSIRT's work concerning if a given asset is or not vulnerable.

PT - If everything is achieved as it is expected with this project, PT will take a big leap regarding the quality of its cybersecurity. In more detail, it will be more aware of possible points of entry regarding vulnerabilities in its ecosystem, and will be able to take quick measures to mitigate the risk of possible exploitation of a vulnerability or even a possible intrusion.

1.4 Document Structure

This report is structured as it follows:

Chapter 2: Vulnerabilities - This chapter intends to introduce a small introduction for what is considered to be the base for this projects, Vulnerabilities. It will be explained what is considered to be a vulnerability, its possible impact, and the motive for discovering vulnerabilities, its lifecycle and so forth.

Chapter 3: VAC - This chapter intends to describe the product of the previous master thesis from were VAC was developed. It will provide a high-level detail of the software.

This chapter is required because there is no other source to get information about what this technology is or how it works, and it is not possible to explain what was done in this thesis without explaining the state of art of this software.

Chapter 4: Design for VACv2 - This chapter contains the original ideas for this master thesis. In other words, it describes what the intentions were for this project to achieve, and the time-schedule for this project.

Chapter 5: Development of VACv2 - This chapter details what was done in VACv2 to achieve the objectives defined in chapter four, and it also illustrates the state of the art for VACv2.

Chapter 6: Evaluation - This chapter compares and presents the differences between VACv2 and the procedures DCY staff had to make to achieve the same result. It also illustrates an example of the events being uploaded by VACv2.

Chapter 7: Conclusion & Future Work - This chapter contains the summary of what was done and achieved with this project. It also presents a few ideas that are possible to explore in order to enhance VACv2 software.

Chapter 2

Vulnerabilities

"Securing a computer system has traditionally been a battle of wits: the penetrator tries to find the holes, and the designer tries to close them." - M. Gosser [10]

Vulnerabilities are emerging at an astonishing pace in this digital era we are living in, and if not adequately addressed it might contain the power to send the world back to the Stone Age.

Vulnerabilities are the foundation of this thesis and are going to be addressed in this chapter. There is no standard definition of what a vulnerability is, but organizations tend to outline it similarly.

In this section, it will be analyzed how world-wide organizations define a vulnerability. It will also address the impact of vulnerabilities and why it is essential to seek for them within the organization's network perimeter, among other features.

2.1 Definition

The word *"vulnerability"* is a noun used when to describe something that is vulnerable. According to the English thesaurus the word "vulnerable" can be described as "*open to assault; difficult to defend*"[11].

Let's have a look at how world-wide agencies, institutes, and organizations define a vulnerability:

CERT

"A vulnerability is a software defect that allows an attacker to violate an explicit (or implicit) security policy to achieve some impact (or consequence)."[3]

Common Vulnerabilities and Exposures (CVE)

"A weakness in the computational logic (e.g., code) found in software and hardware components that, when exploited, results in a negative impact to confidentiality, integrity, or

availability..."[4]

ENISA

"The existence of a weakness, design, or implementation error that can lead to an unexpected, undesirable event [G.11] compromising the security of the computer system, network, application, or protocol involved. (ITSEC)"[6]

IETF - RFC 4949

"A flaw or weakness in a system's design, implementation, or operation and management that could be exploited to violate the system's security policy."[7]

ISO - ISO 27005

"A weakness of an asset or group of assets that can be exploited by one or more threats."[37]

In a nutshell, a vulnerability is a flaw in the hardware or software that when exploited compromises the enterprise's information, thus leading to a violation of one or more of the following properties:

- Confidentiality
- Integrity
- Availability

These three properties, also known as the CIA triad, make a model which was designed to guide policies for information security within an organization.[44]

2.2 Impact of a Vulnerability

According to Intel, the impact of a vulnerability is what "describes the type of harm an attack could cause if the vulnerability were exploited" [8], also they categorize it into three types:

Priviledge Escalation This type refers to vulnerabilities that allow a malicious user to elevate its privileges on a compromised system.

Information Disclosure This type refers to vulnerabilities that if exploited could provide access to classified information.

Denial of service This types refers to vulnerabilities that if exploited could prevent the access of authorized personnel to the system.

It is easy to understand why Intel categorized the impact into three types because each of the types will directly violate one of the properties belonging to the CIA triad. However, and regardless of the vulnerability type or the impact, what will always come out weakened is the company's reputation which is crucial for a successful business. A company's reputation is what provides confidence to customers and stakeholders.

For an organization to gain confidence within its target audience, it is a long and slow process. An attack upon any organization would have an immediate negative effect on its audience's confidence, and to regain such confidence would only get harder. These factors will turn the company's care not to spread bad publicity like the one generated by a successful attack, in others words, organizations have now motivation for finding vulnerabilities within its systems.

2.3 Motivation for finding Vulnerabilities

In May of 2017, the world witnessed a massive cyber attack, probably the biggest one so far, causing an impact at a world-wide scale, it became known as WannaCry[66], and it was a crypto worm. It managed to affect financial services, healthcare systems, telco companies, and so on. The targets were computers running the Microsoft Windows operating system.

WannaCry can be divided into three components, the first focused on exploiting a vulnerability to gain access to the targeted systems, the second was focused on getting Kernel mode for running instructions, in order for the third to cipher the system.

WannaCry compromised the systems through the exploitation of two critical vulnerabilities, one was in the Server Message Block protocol version 1 (SMBv1), which if exploited allowed the execution of arbitrary code in Kernel mode, the National Security Agency of America developed this exploit and named it EternalBlue[65]. The second component also developed by the NSA was known as DoublePulsar[64], and was responsible for implanting a backdoor in the compromised system, while also copying the WannaCry worm into the system and then executing the code.

Companies must prevent at all costs what happened with WannaCry. This event managed to affect the delivery of service by companies, hospitals and more. The WannaCry attack could have been prevented through the installation of the corresponding patches into the vulnerable systems, which were released a couple of months earlier. However, companies are often hesitant to install such patches due to the possibility of it impacting the service.

Organizations also have to be preemptive instead of reactive. There should be periodical scans of their systems to prevent the existence of a vulnerability, but more important to mitigate the risk of exploitation by a malicious user.

2.4 Stages of a Vulnerability

Vulnerabilities exist on account of poor design or development. Nowadays, companies face an aggressive opposition from other companies, and when facing the dilemma of deciding between *"Functionality or security?"*, the second option tends to be left behind most times.

An example of such are release dates, and security is often set on the verge of the release schedule with barely any time to audit the product or make changes regarding the results of such audit. Faster products rather than more secure ones seem to be a vast choice, and everyone loses with this option over the long term.

According to Correia and Sousa (2010) [19] there are three occasions in which a vulnerability could be introduced into the product:

- Designing the solution
- Developing the software
- After the product release

Each one of the stages should not be underrated. Each one of the following examples illustrate a possible decision contributing to a less secure final solution, *e.g.*:

At Design phase If the cipher algorithm is not chosen correctly it could impact the integrity of data managed by the product.

At Development stage When facing a network application if the requests performed by users were not sanitized correctly, it could allow malicious code to be executed.

At Product release If the database original passwords were not changed.

Illustration 2.1 depicts that, in the worst possible scenario, when security is not taken seriously at each of these three phases, the exposure to an attack will increase.

To avoid such a problem, while designing the product, it should be considered possible attack vectors through the identification of hazards and threats. When in development, multiple actions could prevent the introduction of a vulnerability, actions like running a penetration test over the product, or through the use of code analyzers. Finally, when the product is released the risk can be mitigated by placing a firewall, or a WAF (*Web Application Firewall*) between the user and the product, and periodically penetration tests can be done, amongst other options.

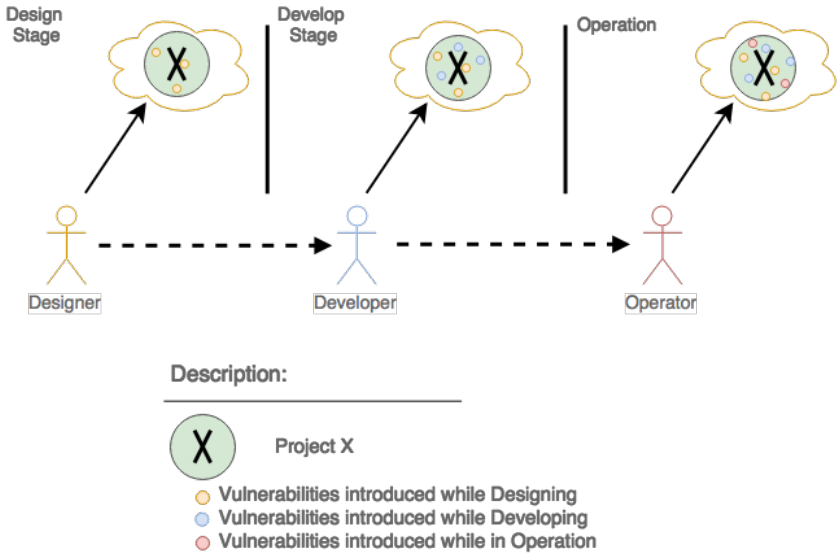


Figure 2.1: Introduction of vulnerabilities at each stage.

2.5 Malicious Software

Malicious Software or *Malware*, consists of any piece of software with the intent of causing harm to a user, computer or network.[1] Malware can take many forms.

2.5.1 Malware Classes

There are multiple types of malware, the following descriptions are according to Cisco Security Research & Operation Sector [15] and to Kaspersky's Labs [39]. The last point mentions more malware types. However, they will not be addressed in further details.

Virus Whenever executed, viruses replicate themselves by inserting its code onto other computer files.

Worms Opposed to viruses, a worm consists of a standalone program with the purpose of replicating itself through the network via the infected machine's network connections.

Trojans Comes from the Ancient Greek story of a deceptive wooden horse that led to the fall of the city of Troy. Trojan's intention is misleading the user from its true objective, in other words, the user is deceived by thinking to be doing one thing but is activating the trojan program.

[1] Defective software is not considered malware, due to its non-intentional damage.

Botnet Cybercriminals breach the security of devices[2] connected to the network, taking control of the infected devices (known as *zombies*) into a network of bots, which are remotely managed by them.

Back Doors A mechanism in which a device's security measures are circumvented inconspicuously into gaining access to the system or its data. Sometimes can be deployed by rootkits, and in other cases, the software manufacturer can leave it forgotten while testing, or also as a method for providing the user a way of restoring his password. Default passwords can be seen as backdoors if not changed.

Exploits Specific software designed to take advantage of a particular vulnerability available in the target, causing unanticipated behavior on the system.

Other Less Common Adware, Drive by Download, Flooders, Keylogger, Macro Viruses, Rootkits, Spammers, Spyware.

2.6 Life-cycle

When a product release occurs, it should not be expected to be flawless. Figure 2.2 helps to understand the life-cycle of a vulnerability in a given product

Having into consideration OWASP's Testing Guide[52], in the first moment, there is what is considered to be a residual risk that increases with time because products are not flawless and it is a matter of time until a vulnerability is found. At this point, there is somewhat as an exponential increase of the risk.

Until a patch is released to fix this problem the vulnerability will become public, and the public will do its job of trying to exploit it, maybe out of curiosity, or it could be with malicious intent. There is a high probability of an exploit for that vulnerability becoming public in a short period of time.

Meanwhile, the risk slowly starts to be mitigated after malware signatures are made available, and even more when the patch is released. Afterward, it is a matter of time until enterprises begin to apply the patch, some may take more time than others, this is an effect of not knowing how it will affect their systems. Therefore it has to be done with baby steps.

After some time, the risk decreases as the patch becomes widely known and a system still vulnerable should be much harder to find.

2.6.1 Patches

As we can infer from the previous section, a patch is a piece of software designed to amend an existing problem, this includes security vulnerabilities, improving usability,

[2]Devices that can range from computers to smartphones or IoT devices (*Internet of Things*).

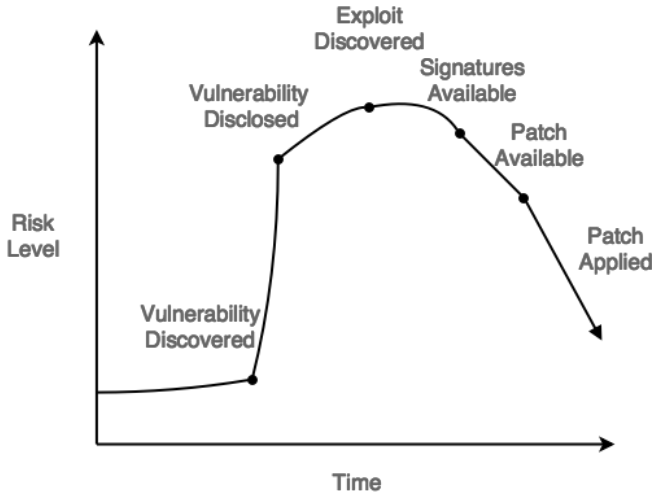


Figure 2.2: Vulnerability Life cycle: Window of Exposure

functionalities or performance[62].

Patches are a quick resolution for a problem, and sometimes they are kept until a new stable software version is released. Although it is meant to repair, as patches are usually a race against time, it is common for them to introduce new problems[61].

2.6.2 Zero-Day

A Zero-Day vulnerability, according to FireEye [29] and Kaspersky [38], consists in an existing vulnerability that is unknown to anyone.

The faster the hacker becomes aware of a vulnerability, more likely he will be successful in the exploitation. Any attacks that try to take advantage of such weakness, at that time are known as Zero-Day Attacks or Zero-Day Exploits.

The Zero-Day is a reference for when the interested parties in charge of such software become aware of the glitch. If these individuals take ten days to discover it, from the hackers perspective, it would be known as a Ten-Day vulnerability.

2.7 Vulnerabilities Categorization

"...In the first half of 2017, Trend Micro's Zero Day Initiative discovered and disclosed 382 new vulnerabilities. Zero-days in 2017 increased to 49 from a mere eight the previous year...." - Trend Micro [48]

Having into consideration the previous citation from Trend Micro, new vulnerabilities are found at a remarkable pace. Illustration 2.3 shows the number of vulnerabilities found each year since 1999[3] until 2017.

[3]The year of 1999 presents vulnerabilities found until that year.

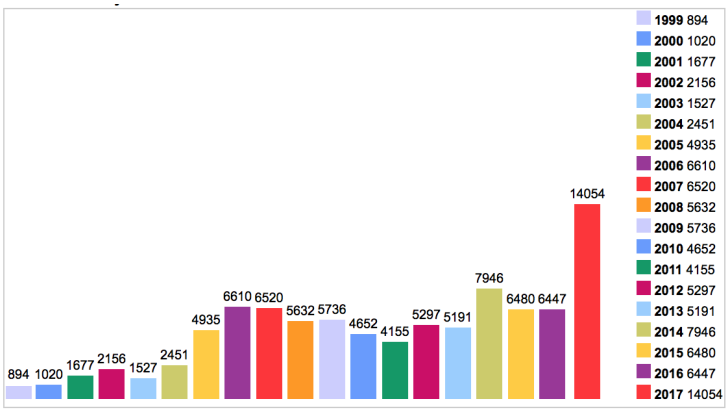


Figure 2.3: Number of Vulnerabilities over the years. [21]

It is easy to realize that there might be a problem for interested parties to become aware of all the publicly disclosed vulnerabilities, and the new ones coming out. To ease the process of spreading the word and managing all vulnerabilities, the Common Vulnerabilities and Exposures was formed.

2.7.1 Common Vulnerabilities and Exposures

Mann and Christey [40] (1999), from the MITRE Corporation, proposed the creation of a shareable database of vulnerabilities. Later that year, Common Vulnerabilities and Exposures, *CVE*, was assembled and 321 vulnerabilities were instantly disclosed[4], and each of these vulnerabilities got associated with a unique CVE identifier[5].

While implementing the CVE List, the responsible team had to improvise a way to group the known vulnerabilities, which evolved into a project of its own called Common Weakness Enumeration (CWE). CWE's[6] purpose is to categorize vulnerabilities by classes, and it works similarly to the CVE List. Both CVE and CWE are own by The MITRE Corporation [63].

2.7.2 Common Vulnerability Scoring System

According to First.org [30], rating vulnerabilities have been a long time challenge, several systems were tried, but none fit the need. Those systems were confusing due to the use of different classifications for the same vulnerability amongst different platforms. The problem was that vendors and stakeholders did not know how to prioritize the threat of the present vulnerabilities.

[4]Remember that in 1999 only 321 were known *vs.* 382 or the first half of 2017.

[5]One CVE represents a cataloged vulnerability. When cataloged it gets an identifier with the following format, CVE-YYYY-NNNN, Y stands for the year the vulnerability was classified, and N stands for its number.

[6]One CWE represents one type of vulnerability. When classified it gets an identifier with the following format, CWE-NNN, N stands for its number.

A universal open standard for scoring vulnerabilities was then designed and is now responsible for grading vulnerabilities, named as Common Vulnerability Scoring System(CVSS). The CVSS consists of three groups, each of these evaluate specific characteristics that depending on the group may be changed.

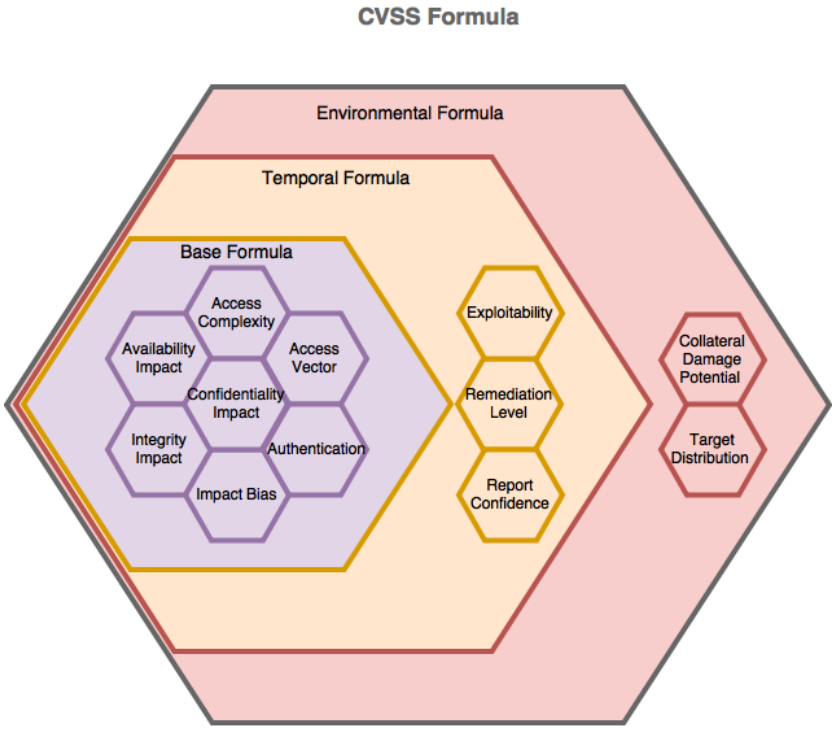


Figure 2.4: CVSS Groups, Metrics and Formula

Illustration 2.4 presents the overall formula, and allows the reader to infer all metrics involved in the calculus of the CVSS, in a more detailed way, it is possible to extract each group's metrics and its dependencies.

2.7.3 Base Metric Group

The base group contains properties that are set by the vendor and are immutable, which means that the result of formula will remain the same throughout time. This group has into consideration the following aspects:

Access Complexity Difficulty of exploring the vulnerability, after the attacker as already gain access to the system.

Access Vector Exploited locally or remotely?

Authentication Authentication required for activating the vulnerability?

Confidentiality/Integrity/Availability Impact Information delivery affected?

Impact Bias Allows to assign more weight to one of the CIA properties over the remaining.

While implementing this project, it has only been emphasized this group before the remaining ones. The reason is because PT has several complex systems that may change at any given time, making difficult to operate over the remaining metrics groups that might vary on these conditions, making this group more suitable.

2.7.4 Temporal Metric Group

The temporal group is responsible for metrics throughout the life of a vulnerability. Its metrics are set by the vendor and are mutable. This group relies on the base group to be computed, and has into consideration the following aspects:

Exploitability Code or exploits available?

Remediation Level Workarounds or patches provided by the vendor?

Report Confidence Degree of confidence in which the vulnerability is real.

2.7.5 Environmental Metric Group

The environment group measures the effect that a vulnerability might have in its vicinity. Its metrics are set by the end-users and are mutable. This group depends on the temporal group for being computed, and has into consideration the following aspects:

Collateral Damage Potential Physical damage or loss possible?

Target Distribution Susceptible surrounding environment where the vulnerability was exploited?

2.8 Conclusion

Vulnerabilities are the key for this project, and the motive for finding vulnerabilities or the impact they can have in the system is the base for this master thesis. This chapter tries to explain what a vulnerability is, also described some concerns related to vulnerabilities, how they are categorized and more important, prioritized.

The most important thing to retain of this chapter is that a vulnerability is a flaw in the hardware or software that when exploited compromises data, and if exploited it can cause irreversible damage to the organization.

Chapter 3

Portugal Telecom's Vulnerability Assessment

"Companies spend millions of dollars on firewalls, encryption and secure access devices, and it's money wasted, because none of these measures address the weakest link in the security chain." - Kevin Mitnick, Security Consultant

"There are only two types of companies: Those that have been hacked and those that will be hacked." - Robert S. Mueller III, former FBI Director

In the last chapter, we have seen the basis of this essay, vulnerabilities. We saw what defines a vulnerability, their lifetime, the impact they might have, how they are scored, and more.

In this chapter, we will analyze the case of Portugal Telecom regarding the procedures for discovering and managing vulnerabilities. It was problems concerning these procedures that lead to the development of a custom software named Vulnerability Assessment Coordinator (VAC). VAC software aimed for a centralized and automized way of managing vulnerability scans over Portugal Telecom's network infrastructure.

In this chapter, we will examine the original problem of Portugal Telecom. Then we will see what technologies the organization used at the time. Subsequently, we will get into VAC software in more detail, specifically its architecture, and the purpose of its main components. Finally, we will get to the pros and cons of VAC, which lead to the Continuous Security Assessment master thesis.

3.1 DCY's Lacuna

Continuous Security Assessment thesis is a significant improvement of a previous project named Vulnerability Assessment Coordinator. VAC was originated in a joint venture between the Universidade de Lisboa, Faculdade de Ciências and Portugal Telecom, which resulted in a master thesis.

The intended solution was a mean to manage vulnerabilities in the most automatized way possible. This because the process of detection and mitigation of vulnerabilities for assets exposed to the internet was done manually, and in a nonregular manner.

3.1.1 Vulnerability Scanning

NIST defines the term vulnerability scanning as being what "... identifies hosts and host attributes ... but it also attempts to identify vulnerabilities rather than relying on human interpretation of the scanning results."[51]

Typically, a vulnerability scan is composed of two main phases: Discovery and Scanning. The discovery phase is about making a reconnaissance of the target, in other words, it is about discovering open ports, identifying services, inferring the operating system, and so on. The scanning phase compares the responses provided by the target for matches against known vulnerabilities.

The setup of the scanner technology's appliances is actively involved with the type of intelligence we pretend to extract from the vulnerability scans. Two crucial decisions have to be made regarding a scan. The first is while setting up the scanning appliance, while the second is when making the scan configuration.

The first decision is about the view of the scanning appliance. When studying the place where to set the scanner, there two options each with advantages and disadvantages.

External We would get a better understanding of the challenges imposed to an adversary while trying to get a possible access point. However, the results related to the asset might not be accurate, due to the possibility of having defense mechanisms protecting the target, *e.g.*, Firewall, WAF.

Internal A local scan would indeed provide vulnerabilities with a higher percentage of confidence. However, we would not have the adversaries point of view to access the system. This perspective is valuable because we should take into consideration if the weakness is or is not in range of the adversary, *e.g.*, with the use of a firewall, a higher rank vulnerability could be overcome by another vulnerability less critical.

Concerning the authentication of scans, it should be seen as an improvement on the confidence level of the reported vulnerabilities. What does this means, both scan types compare the obtained responses from the target against a vulnerability database for a match. The difference is while a non-authenticated scan might identify a given service with a version that is known for being vulnerable, leading to the signaling of that vulnerability in the report. The authenticated scan could verify if the asset contained some sort of patch fixing that vulnerability, meaning that there is no vulnerability.

Another factor is that every scanner engine is different and contain different purposes, *e.g.*, some are more web-orientated, others more services orientated, and so forth. The

use of a single vulnerability scanner is not full-proof. However, the use of more than one can increase the odds in our favor.

The problem here is, there are no standard metrics for vendors, despite what we have seen in the last chapter, vulnerabilities have CVSS managed by a third party, but other parameters are not. Other parameters considered by vendors have the purpose of helping customers prioritizing weaknesses, but they have no common ground, *e.g.*, the severity field that almost every scanner accounts. One scanner might value it from one to five, while another may value it as low, medium, or high.

Also, most engines are not cloud-based. Fact is most internal scanners have to be managed by customers, and it might require human interaction before a scan to perform operations like updating the operating system, updating the appliances' vulnerability database, and so forth. These factors have to be taken into consideration since it might take much time to do before launching a scan.

Vulnerability scanners have a feature that is essential to mention, it is called template, and consists of the guidelines for running the scan to the target. The vulnerability scanners will only do what is specified in the selected template, *e.g.*, if we want a TCP port scan, it must be stated in the scanning template.

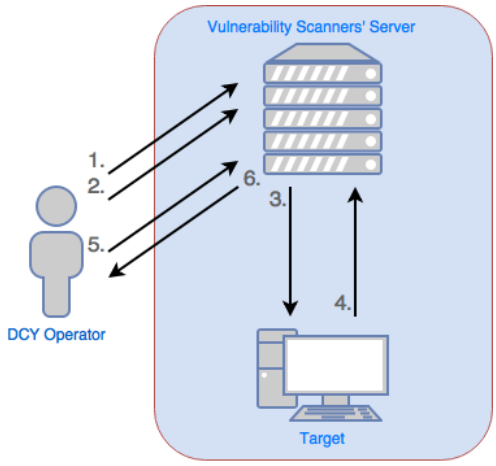


Figure 3.1: Launch Vulnerability Scan Procedure

Image 3.1 illustrates the launch scan procedure handled by a DCY operator. The numbers contained in the illustration are ordered by execution, except for steps three and four that are exemplifying the communication between the scanner and the target. Next, each number is matched with the correspondent task.

1. Update the server's OS and appliance's vulnerability database;

2. Configure the target and schedule the scan;

3. Probes ports and requests are made from the scanner;

4. Responses provided by the target;

5. Request the vulnerability technology scanner for the generation of the scan report;

6. Vulnerability report provided.

All these tasks result in a significant setback for PT personal.

3.1.2 Vulnerabilities Reporting

Parsing all the information provided by the vulnerability scanning reports can prove to be a laborious task.

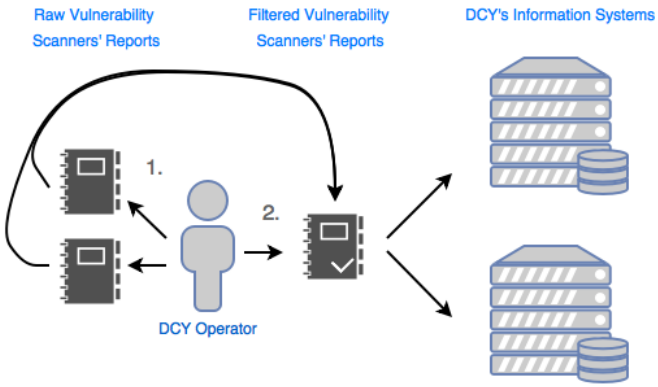


Figure 3.2: Upload Vulnerability Reports Information

The procedure associated with the upload of the scan report results to DCY's repositories performed by a DCY operator are exemplified in image 3.2. Next, each number contained in the illustration will now be matched with the correspondent task.

1. The DCY operator analyses the erudition provided by each report[1], correlates, and filters it by "making a new report" with the essential data.

2. With the filtered report, he starts uploading it to all of DCY's information repositories.

All the necessary tasks in the previous section plus these ones represent a vital impact on DCY's staff that already have much to take their time.

3.2 Technologies in use at 2015/2016

DCY used to work with:

[1]PT contains more than one scanner technology performing scans to their assets.

- Vulnerability engines:
  - Greenbone Networks' OpenVAS
  - Rapid7's nexpose
- Information repositories:
  - ArcSight ESM
  - High-Performance Infrastructure for Data Research (Hidra).

3.2.1 Vulnerability Scanners

Let's have a look at the vulnerability scanners used by DCY at the time.

OpenVAS Open Vulnerability Assessment System is a Free to Use framework that aims to provide a full vulnerability management solution. OpenVAS is structured in three modules. They are named Clients, Services, and Data. [32]

The Clients' module is responsible for handling clients. It provides two way to interact, one via Web Service and the other via Shell. The Services' module, is where the magic happens, this is the place where the scanner and the manager are located. The manager is where all the intelligence of the OpenVAS is, it controls the scanner, and interacts with the Clients' and Data's modules. The Data module contains the Network Vulnerability Tests' (NVT) and communicates strictly with the Service module. The NVT's contain the tests to be executed by the scanner and updated on a daily basis, it can be updated from a commercial feed service or by OpenVAS NVT Feed, which is free. The Data's module also contains a database where all the pasts results and configurations stay.

One significant advantage of OpenVAS, aside from being free, is the fact that it could be used in a master-slave config, being the master responsible for updating the slaves, and also scheduling their tasks in the best possible way.

Nexpose In reverse to OpenVAS, Nexpose is subscription based software. Rapid7 is Nexpose's manufacturer and offers two vulnerability scanning solutions. The first is InsightVM which is a suite with all their tools for vulnerability management, and secondly, Nexpose which they refer to as being an on-premise vulnerability scanner solution. [58]

Nexpose being a proprietary software besides from the server is deployed at, is a self-sustained software, meaning Rapid7 manages its vulnerability database updates.

A few advantages that Nexpose can count on is that every vulnerability validation is integrated with Metasploit, another tool belonging to Rapid7, another is that Nexpose keeps up with the vulnerabilities found in the systems from the moment it is detected until the removal, among several others.

The problem with these scanners is that both require workforce from DCY. From the server's update to the target management, or the scan scheduling to the report validation, all these chores take precious time to DCY personal.

3.2.2 Information Systems

The information systems at the disposal of DCY for storing asset's vulnerabilities at the time were two:

ArcSight ESM ArcSight ESM is a powerful and efficient SIEM (Security Information and Event Management) software. INFOSEC Institute describes a SIEM as being "... a software solution that normalizes, filters, correlates, assembles, and centrally manages other operational events to monitor, alert on, respond to, analyze, audit, and manage security and compliance pertinent information. SIEM systems provide fundamental security operations like other product categories."[35]

This technology is licensed and was acquired by PT. Like the description stated, SIEMs correlate information from multiple sources and allow Security Operation Center (SOC) teams to act more efficiently. In the context of the project, ArcSight collects information on the targets from the vulnerability scanners.

Hidra Is an in-house development of PT, and it is composed of three software RabbitMQ[55], ElasticSearch[25], and Kibana[26].

RabbitMQ is an open source message broker, used for its advance message queueing protocol, which allows a reliable change of vast quantities of events.

ElasticSearch is a distributed, RESTful search and analytics engine capable of handling with massive amounts of information. Although the data managed by ElasticSearch is centrally stored, we can query the data and expect quick results smoothly with the help of JSON[9].

Kibana is a visualizer for the data stored in ElasticSearch. It allows to graphically represent the information efficiently and quickly through the use of custom dashboards that aggregate, and filter the data stored. Both Kibana and ElasticSearch are from the same manufacturer[24].

From these components, it is possible to understand that Hidra was aimed to be a tool for handling a massive quantity of data and allow a rapid visualization of that intelligence.

3.3 VAC Architecture

As we have seen in the previous sections, vulnerability detection and mitigation are crucial when to secure the ecosystem of any company, and PT has to spend much time to assure this. However, DCY has not much time to spend on these tasks, becoming the second choice against more critical situations. From this insufficiency, the idea for an automatized vulnerability assessment and management was born.

The Vulnerability Assessment Coordinator is the project that would come to solve DCY's demand. It was meant to be a centralized platform where it would be possible to:

- Upload the configuration for new scanner engines[2].
- Configure new assets (IPs, IP Ranges, or Hostnames).
- Schedule scans[3].
- After the first scan[4], it would be possible to flag vulnerability events as exceptions, which meant that from that time forward it would be regarded as a false positive.

Image 3.3 illustrates VAC's principal components, and how they interact with the vulnerability engines and the information repositories.

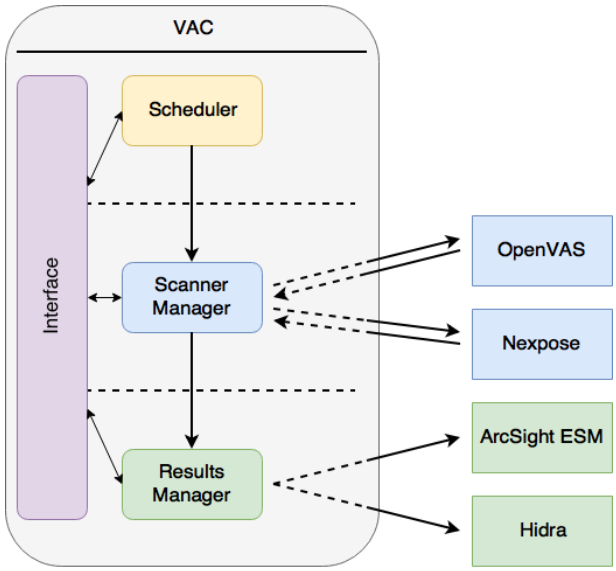


Figure 3.3: VAC Architecture

After looking at the previous image, we can, without doubt, see four specific modules, let's now have a closer look at each one and analyze how they interact with its vicinity in a more detailed manner.

[2]However, the only technology supported was the ones that PT possessed at that time.
[3]The results of the scans would be automatically uploaded to the data repositories.
[4]The term *First Scan* refers to the first time an asset is scanned through the use of VAC, and not VAC's first scan ever.

3.3.1 Scheduler Module

The scheduler module is responsible for keeping every scan configuration, and it uses an internal object named *scan* to interact with other modules. The scan object was designed to have two subtypes, which are:

Spontaneous Scan Is a scan that only has one occurrence. Also, it is to occur as soon as possible and provides the user the power to select what scanner engine to perform the scan.

Periodic Scan Is a scan that has a frequency property associated, which makes it useful for launching a scan at a specific date and time, *e.g.*, when it causes less disruption to the asset delivering service.

Now that we have seen the types of scan, let's analyze de scan object. The scan object *per se* is not only where all the information related to the targeted assets is stored, but also where the scan configuration is kept. When the operator is defining the object, he sets the following properties:

ID Scan Identifier

Template VAC template[5] name, this represents the guidelines for the scanning itself.

Hosts The assets to be scanned, as mentioned previously, this could be IPs, IP Ranges, or Hostnames.

Start The date and time for the scan to be executed.

Available Period This is a concept for the scan to be paused or resumed if a specific time is reached, *e.g.*, we pretend that a scan does not disturb the regular working hours of a targeted asset, so the available period to start could be from 8 pm until 6 am.

Options This is a key-value dictionary. Besides others values, this is where is stated the criticalness of the scan.

Scanner[6] Is a property that allows the operator to choose what scanner engine technology to be used while scanning the target.

Frequency[7] Is a property that can range from: Once, Daily, Weekly, or Monthly. It allows to schedule a scan to occur at a time in a periodical manner.

In the periodic scan type, the scanner technology is selected according to the criticality option. If defined and equal or higher to seven, then the primary scanner[8] is selected.

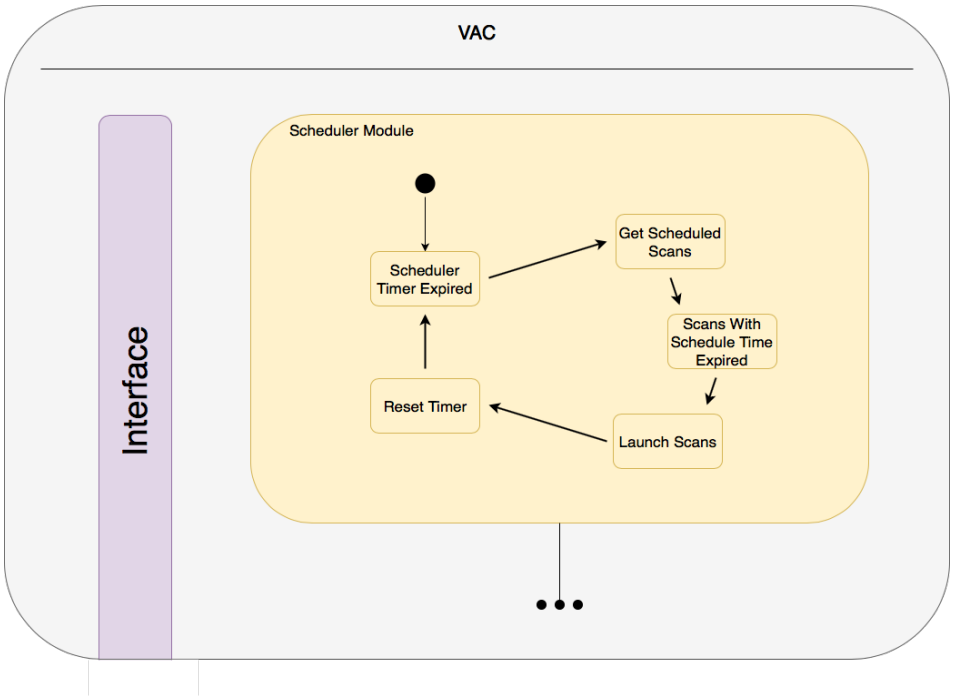


Figure 3.4: Scheduler Module Steps

Illustration 3.4 helps to understand the steps of the scheduler module, and an explanation of what it has been done is provided after.

So as said before, the scheduler module maintains every scan configuration in its possession. At its core, there are two central properties, a timer and a concept of a primary scanner technology.

The purpose of the timer is for checking if any scan has to be executed. If this is the case, the scan configuration is sent to the scanner manager module to perform the scan, and the timer is reset. As to the primary scanner attribute, DCY between OpenVAS and Nexpose, saw it fit to assign Nexpose more significant tasks, becoming Nexpose responsible for scanning more critical assets, due to more reliable results. What this tries to achieve is a means for maximizing the scanner appliances' effectiveness, which is done by assigning scans of less critical assets to the not primary scanners, and scanning the truly critical assets with the primary scanners.

3.3.2 Scanner Manager Module

In a similar way to what was done in the previous section, we will first address the objects that the Scanner Manager module interacts with, and from that, move onto the scanner manager analysis *per se*.

[5]VAC Template is an object from the Scanner Manager Module, and will be addressed ahead.
[8]The primary scanner is a property of the Scheduler Module, and will be addressed ahead.

This module interacts with two objects besides the scan object, which names are *scanner* and *VAC template*.

The scanner object represents one instance of a scanner engine technology, through the retention of information regarding that instance. Its attributes are:

ID The scanner instance identifier to VAC.

Host The address where this scanner is located can be an IP or Hostname.

Scanner Type Scanner Engine Technology, which is for this instance to represent. Only can be OpenVAS or Nexpose.

Credentials The username & password for VAC to log into the scanner appliance instance.

Configurations This is a key-value dictionary. It is possible for VAC to define some extra configurations, *e.g.*, the maximum number of scans that can be happening at the same time, among others.

The Scanner Manager Module is able to establish a connection with the scanning appliances through the use of application programming interface (API) made available by the vendors. When the connection is active, and the scanner manager authenticates successfully, it can perform operations on the appliance.

In the last section we saw that when defining a scan object, one of its properties was a template. This property identifies which VAC Template object is selected when preparing the scan. The VAC template object is a mere association between all equivalent vulnerability engine's templates and VAC. The following example will try to help to make it more clear, *e.g.*, imagine in each scanner technology a template that will only perform a scan to TCP ports, so we take each template's name from each scanner technology and create a VAC Template with the templates' names there. When defining the Template in VAC, we face the following properties:

ID The template identifier to VAC.

Scanner Templates It can be seen as a key-value dictionary, where the key would be the type of scanner technology, and the value would match to the name of the template for that same technology.

The scan object is also associated with the Scanner Manager module, which interaction starts at the Scheduler Module. When the scan information switches to this module's possession, it sees its responsibilities grow as it makes part of this module to supervise every scan occurring.

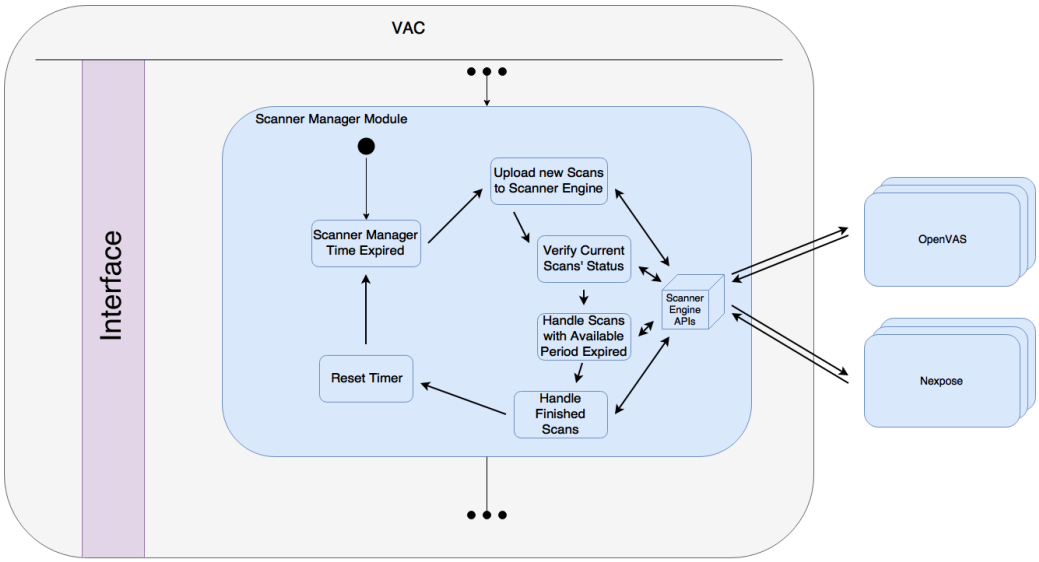


Figure 3.5: Scanner Manager Module Steps

Figure 3.5 is trying to illustrate that after the scheduler module asks for a scan launch, the scanner manager module will become responsible for the scan. Similarly to the last module, this module also has in its properties a timer. The timer's purpose is to periodically trigger the Scanner Manager module's routine of verifying new scans to launch and to monitor the active scans. When a scan changes its state, this module will forward it to the accountable module. The "turn" ends when the Scanner Manager module finishes checking all scans at its possession, at that time the timer is reset.

3.3.3 Results Manager Module

The Results Manager module interacts with two other objects besides the scan object, whom names are *processor* and *exception*.

The processor object will handle a given scanning technology report. Every scan report issued contains a specific structure depending on the issuing scanning technology. In a nutshell, VAC will have to contain as many processor classes as there are scanning technologies[9]. The processor objects will be the ones capable of handling the scanning technologies reports because they are the ones knowing the scanning technology report's structure. Also, they contain every DCY's information repository's connector within itself, in order to connect and upload the results to the DCY's repositories.

Image 3.6 tries to ease the understanding of the Results Manager module by presenting how it is processed. The scanning technology's reports will be the input of the corresponding scanning technology processor, and then the processor will handle the reports, and diffuse the results by the DCY's repositories.

[9]Which ultimately will only exist two, one for OpenVAS and the other for Nexpose.

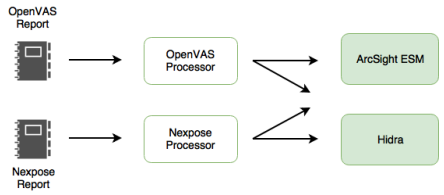


Figure 3.6: Scheduler Module Steps

The other object that interacts with the Results Manager module is the exception object. After a scan occurrence, the results are uploaded to the information repositories of DCY. At this point, the DCY operator will perform an analysis of the results obtained, and if the existence of a false positive is verified, then that particular event will have to be excepted from future scans, this is done through the use of an exception object. This object is composed of the following attributes:

Vulnerability Name The vulnerability name in the scanning technology.

Host he host where the vulnerability event was signaled.

Port Protocol The vulnerable protocol.

Port Number The vulnerable port.

Scanner Type The scanning technology name that signaled the vulnerability.

Person The person who vouched for the exception.

Reason Motive why it is to except.

Result This field is a copy of the result field that makes part of the vulnerability classification in the scanning technology.

When a scan finishes executing, and this module executes its procedure by comparing every vulnerability event against its exception records. If a match is found, then the vulnerability event would not be uploaded to the DCY's information repositories.

The Results Manager module will hold on to every scanning technology processor and every existing exception.

Image 3.7 illustrates the regular procedure of the Results Manager module[10]. Similarly to the previous modules, it makes use of a timer, but it also has a repository in the server where the solution is deployed. The timer attribute in resemblance to the other modules, allows this module to check if finished scans were received. Afterward, it will

[10]Being the Scanner Manager module (SMM) responsible for all interactions with the scanning appliances, the SMM API represents the pointer that the Results Manager module contains for being able to request services to the scanning technologies.

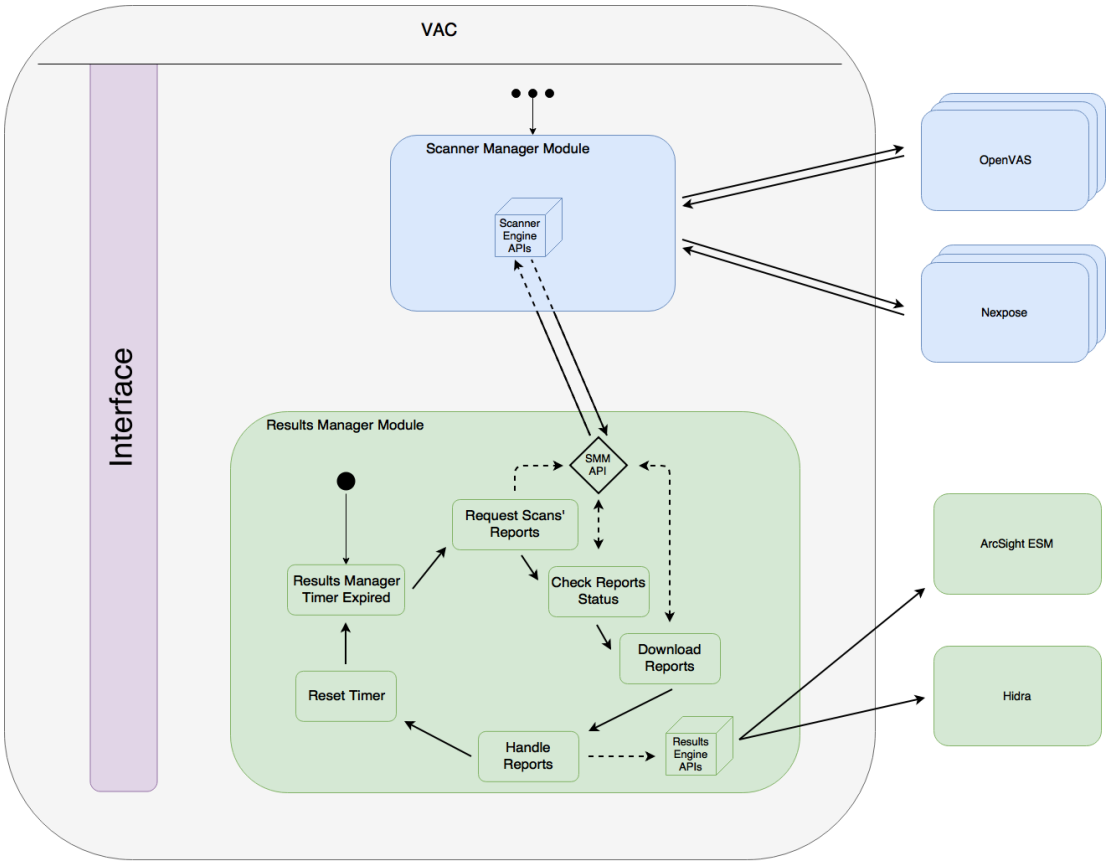


Figure 3.7: Results Module Steps

request their reports, monitor its status, and download the report when it finishes. Then it will send it to the responsible technology processor.

The other attribute is a repository, which purpose is mainly for logging. When the processors finish the handle and upload of the results into the DCY's repositories executing, it creates an empty file related to then scan that got processed. This feature allows the DCY operator, to check if an error occurred due to the presence or not of the file.

3.3.4 Interface Module

Last but not least, we now are going to have a walkthrough over the Interface Module. This module is responsible for handling the view layer of the VAC project. It is what allows the DCY operator to communicate with VAC itself. VAC can be only operated through the use of this module.

In figure 3.3, it was represented interactions coming and going from this module, the ones starting on this modules are mainly setter operations, while the communications ending on this module are getter operations.

Image 3.8 shows the first page when accessing VAC. This page will show scans occurring at the time we access the VAC website. Due to the size of the images, other

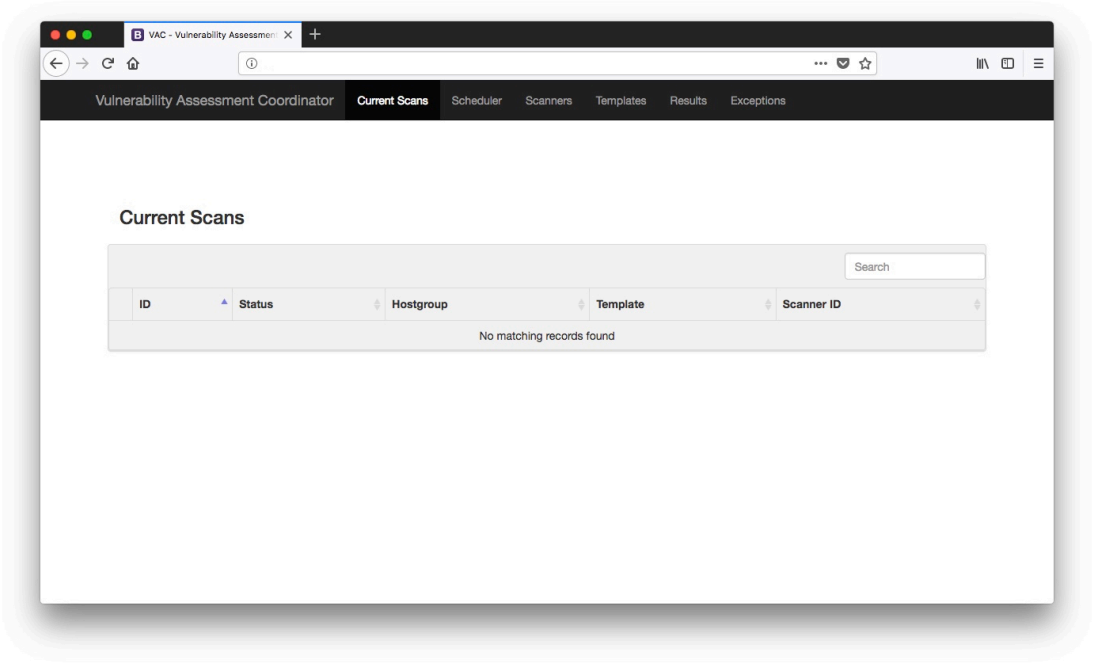


Figure 3.8: Welcome Page

views from VAC will be available in Appendix A section. However, it will be provided descriptions of those same views ahead, and what we can accomplish with them.[11]

A.1 - Scheduler View On this screen, it is possible for the user to check all periodic scans configured, he can also remove them. As to the spontaneous scans, it has the same behavior, however, when a spontaneous scan finishes it is automatically removed from the list.

A.2 - Scheduler Configuration's Properties On this screen, it is possible to watch the current values of the attributes referred at the end of section 3.3.1 Scheduler Module, and also to change them.

A.3 - Configure New Periodic Scan[12] On this screen, it is possible to configure a new periodic scan through the fulfillment of the attributes of a scan referenced in section 3.3.1 Scheduler Module.

A.4 - Scanner View On this screen, it is possible to check the current scanner appliances for each scanning vulnerability technology, the operator can also remove them.[13]

A.5 - Add Scanner Configuration On this screen, the operator can add a new scanner

[11] All the views retracted are related to the previous modules. They will present how the operator can interact with VAC, while also trying to ease the understanding of the previous sections.

[13] The Scanner Configuration's Properties view is not shown because it is equal to the Scheduler Configuration's Properties but only contains the timer box.

appliance through the fulfillment of the attributes of a scanner object referenced in section 3.3.2 Scanner Management Module.

A.6 - Template View (Expanded) On this screen, it is possible to check the current templates[14], the operator can also remove them.

A.7 - Configure New Template On this screen, the operator can create a new VAC Template object. Note that the "Scanner Type" field takes the scanning technology name, and when the "New Configuration" button is pressed, a text form pops up and allows the insertion of the template name that matches that scanning technology.

A.8 - Processor View On this screen, the operator can see the processors that are currently working, and he can also remove them.[15] The processor is one of the objects described in section 3.3.3. Results Manager Module.

A.9 - Results Manager Configuration's Properties On this screen, besides the timer box it is possible to observe that the operator can decide where the results repository will be. The repository is what was referenced in section 3.3.3. Results Manager Module, as being useful for logging operations.

A.10 - Exception View (Expanded) On this screen, it is possible to get the list of current exceptions and their information to the target asset. The operator can also remove them at any time.

A.11 - Configure New Exception On this screen, it is possible to configure a new exception by fulfilling the fields as they were in the report.

3.4 VAC - Pros & Cons

The original objectives for VAC were:

1. Automatical scan configuration across the available scanning technologies.
2. Automatical periodical scan to PT's infrastructure.
3. Automatical upload of the results to DCY's information repositories.

These objectives were achieved, and the development of the solution came to facilitate the work of DCY's personal. It allowed them to cease doing repetitive chores like the scheduling of scans on the scanner's appliances scattered along PT's ecosystem, or

[14]This was referenced in section 3.3.3. Results Manager Module as the VAC Template object.

[15]The Add Processor view is not shown because it only presents a text form to insert the name of the processor class.

the manual process of verifying the scan reports or uploading the information into the repositories.

However, due to time limitations, VAC was like a "Diamond in the rough", being a tool with tremendous potential but also demanding the operator to know how the software runs. VAC was also restricted due to some limitations of the scanning technologies, but the primary factor was that VAC was not designed to be scalable regarding any technologies, in other words, VAC could not be expanded to work with any other technology without a significant development.

Concerning the technology deficiencies, VAC will only be able to perform vulnerability scans, which identify and detect vulnerabilities related to flawed software or misconfigured assets. Also, taking into consideration that Nexpose's scan reports are more empowered information wise, VAC was designed to convert OpenVAS' scan reports to the format of Nexpose's reports structure, becoming the canonical format for the upload of the results to the repositories.

Another topic is precisely the information repositories, which the operator has no control over them. PT contains clients whom may have bought a suite of services, and vulnerability scanning may be a part of this suite. With VAC, the upload of the results to the repositories is automatical, and in this case, it is unwanted behavior. Another factor is, there are only two technologies available with no possible way to expand or remove them.

3.5 Conclusion

PT was facing a concern that consisted of having to spend much time with vulnerability scanning related tasks, which most times had to be set aside given the duration of such tasks and the appearance of more urging matters. Having this problem in hands, it was drawn the guidelines for an automatical system for vulnerability scanning.

A partnership between PT and FCUL lead to a master thesis regarding the previous problem. The Vulnerability Assessment Coordinator was the outcome of this collaboration. The project aimed for a reduction of tasks related to vulnerability scanning by DCY's personnel, through the automation of periodical vulnerability scans over PT's assets and the upload of the scan's results into DCY's information repositories.

DCY's tasks were reduced to the maintenance of the servers where the vulnerability scanners were deployed in, and the post-processing of the information uploaded to the repositories. However, VAC got stuck to the technologies PT had at the time, becoming a non-scalable system.

Chapter 4

The Project of Continuous Security Assessment

"As we've come to realize, the idea that security starts and ends with the purchase of a prepackaged firewall is simply misguided." - Art Wittmann, Freelance journalist and writter specialiazed in IT.

"In the very near future, cyber security exercises are going to be absolutely expected of all companies by regulators." - Michael A. Vatis, partner in the New York office of Steptoe.

In the last chapter, we witnessed how PT tried to counteract the vulnerability assessment procedures, which was already a tremendous time-consuming task even if concerning critical assets only. The solution was the creation of a software, VAC, which purpose was to reduce the human interaction on the vulnerability assessment to a minimum. The procedures that kept requiring intervention were operations like, *e.g.*, maintenance of the servers where the vulnerability scanners were deployed, or the analysis of the results uploaded into the DCY's information repositories.

This chapter will present the limitations of VAC, which lead to the guidelines that were behind VACv2. It will also describe what was intended for VACv2 to accomplish. Last, it will analyze the original planning of this master thesis and compare it to the real schedule with the reasons that cause the project to drift.

This chapter will first present VAC limitations and the VACv2 main objectives. At this point, we will get into more detail of each objective to understand what is pretended from each one. The following section will be presenting the original planning *vs.* the real planning with the causes why the original deadline drifted.

4.1 The Objective of the Project

The name of this project was not randomly chosen, and it is intended that at the end of this project a system is built which will continuously perform security assessments of PT's platforms.

VAC was unable to achieve its purpose mainly for two reasons, first was the difficulty in using the software. However, with time this could have been overcome, the problem was that VAC got short on time due to PT ceasing its contract with Rapid7 regarding Nexpose, and started to use Qualys cloud-based vulnerability scanner which VAC was not able to integrate, and so a need for a VACv2 was born.

To perform vulnerability assessments continuously while trying to make it more autonomous is the ultimate goal of this project. To achieve this end, and to avoid losing the effort spent in VAC, the objectives for VACv2 were outlined:

Improvement of VAC Overall enhancement of the tool, making its scalable regarding technologies, improving usability, and protecting its data from unwanted access;

Addition of a new Scanning Technology into VACv2 addition of the new scanning technology, Qualys, and meeting its requirements;

Results treating & Data correlation - Improving the results manager module and providing control to the operator.

Let's now approach what is intended from each objective presented above.

4.1.1 Improvement of VAC

VAC was originated from a master thesis. Nonetheless, it was a project with effort and costs associated, and PT did not saw it lightly just to set it aside due to its emerging limitations. The first objective defined for this thesis was the improvement of VAC, which lead to the development of a new version, hence the name VACv2.

By the time this thesis started, DCY already had assessed the tool for its deficiencies and possible improvements. These requirements are stated ahead, and afterward, they will be approached in further detail.

- Automatical Target Identification
- Authentication of Users
- Authorized Users (Whitelisting)
- Cipher Sensitive Data
- Cookie-based Session

- Correlation Data Integrators[1]
- Creation and Management of Mailing Lists
- Data Pre-Load
- Editable Configuration Files
- Improvement of Data Management (Creation, Update, and Removal)
- Improvement of Logging
- Improvement of Scan Recurrence
- Notifications & Custom Mailing
- Technology Loading

These requirements will provide changes both in server side as the client side of the solution. While developing the solution the student was allowed to add any new features that could improve the software.

Let's now analyze what is pretended from each of these requirements.

Automatical Target Identification

This was a highly requested feature by PT, and perhaps, the most influential feature in VACv2 because of the changes it requires over the code of the scanner appliances and the scan configuration.

For this feature to be accomplished, it requires that the solution to be aware of what network an appliance is responsible for scanning. It is intended to VACv2 to automatically assign the correspondent appliances to the targets while the operator is making the scan configuration.

In the end, this feature would try to map a given target with the correspondent network/ appliance autonomously.

Authentication of Users

The data handled by the solution is sensitive, not only because it handles confirmed vulnerabilities, but also because those vulnerabilities were reported in high-valued assets belonging to PT. VAC had no authentication, meaning that anyone could reach this information, becoming itself a liability.

This mechanism is intended to work with resort to Active Directory Kerberos authentication, meaning that users credentials will have to be exchanged with the server. To

[1]This feature fits into two objectives: Improvement of VAC; and Results Treating & Data Correlation

securely exchange credentials with VACv2, it means that the communication between the client and server has to be secure through the use of a certificate generated by PT.

In the end, it is supposed that this feature adds a secure communication between the client and server, and Active Directory Kerberos Authentication of the user, making sure only PT personal will access the information.

Authorized Users (Whitelisting)

This feature aims to restrict access to VACv2, meaning that only authorized users would be able to access the solution. This feature will work based on a whitelisting technique for users.

In the end, this feature combined with the Authentication of Users presented previously will guarantee that only the necessary PT's personal identified in the whitelist will gain access to VACv2.

Cipher Sensitive Data

We have already witnessed the addition of two security layers over the solution. However, these measures would only apply to users trying to access the client-side of VACv2. An example of data that VACv2 needs to store is when configuring a new scanner appliance, the credentials for accessing such appliance must be stored by the software.

This last example tries to make aware that is not only the communication client-server or the access to the solution that needs security, but also the back-end information. All the information stored on the server that is considered sensitive must be protected, and this is done with resort to ciphers.

However, another feature that will be mentioned ahead is the possibility of having editable configuration files, this means that only data like credentials and such properties will be ciphered.

Cookie-based Session

This feature aims to associate a user with a session. The session will be assigned to a user after a successful authentication into the platform and will change periodically. The periodical change of the cookie will try to guarantee that the session is not hijacked.

This feature combined with the Authentication of Users & Authorized Users (Whitelisting) will provide a more comfortable security layer over VACv2.

Correlation Data Integrators

In VAC, it was not convenient to launch scans targeting clients because their results would be automatically uploaded into DCY's Information Repositories, which is supposed to maintain only PT's data.

This feature's objective is to provide the operator whether if it is to upload the data to an internal Information Repository, and in that case which one of them, being also possible to choose none or all of them.

In the end, VACv2 will allow the operator to choose what to do with the scan results.

Creation and Management of Mailing Lists

VAC already handled emails. However, this feature was not the most user-friendly. The operator needed to have an in-depth knowledge of how the software to use this feature and would need to specify the emails to notify in every scan configuration.

This feature aims to provide a way for the operator to create and manage mailing lists which will contain emails to be notified regarding an action. In the end, this feature will allow the operator to specify a mailing list to notify and what should trigger the notification, *e.g.*, starting or ending of a scan.

Data Pre-Load

This feature is easier to explain with resort to an example. Let's recollect the configuration of VAC concerning templates, where the operator had to configure every one of the templates he wanted to use. This is not very practical or user-friendly.

What this feature aims to achieve is the loading of the available information into every property on the client-side, every property where this feature makes sense. For example, when the user is configuring a new scan, VACv2 should automatically fetch all the available templates regardless of the scanning technology.

Editable Configuration Files

DCY's cybersecurity engineering team, which is the team responsible for handling this tool had a specific request. They pointed out that in some cases VACv2 might be easier to handle through the use of editable files than the interface of the application *per se*.

They requested for the configuration files to become editable in VACv2, something that did not occur in VAC.

Improvement of Data Management (Creation, Update, and Removal)

VAC could not handle mistakes, the way the operator had to solve them was through the removal and configuration of a new object.

This feature's purpose is to make every configuration in VACv2 ready for editing, avoiding to give the operator more work than is needed, *e.g.*, there are scans which targets could reach hundreds, and if a mistake were made in these configurations, it should be possible to edit *vs.* having to create a new object.

Improvement of Logging

VAC as any other software, already provided logging. However, it was somewhat generic, not helping when a problem occurred. Also, VAC store big log files which could be confusing and information valueless.

It is intended for VACv2 to provide more detailed and intuitive logging, renew on a daily basis.

Improvement of Scan Recurrence

VAC already handles scan recurrence. However, the new scanning technology requires it to become more generic. The way VAC handled recurrence was allowing the operator to set a value between daily, weekly or monthly, which had a few restrictions, *e.g.*:

- For daily scans, would run every day at the same time.
- For weekly scans, would run accordingly to the day chosen, which means that if the scan was meant to start at the tenth of the month, and the tenth day was a Monday, then every recurrence of that scan instance would run on a Monday.
- For monthly scans, continuing with the example provided for the weekly scan, every month scan would run on the tenth of each month.

However, the new technology provides more power concerning the options available, *e.g.*, it is possible to run scans every third Thursday of a month. In the end, VACv2 is intended to support this feature from the new scanning technology acquired by PT.

In the end, VACv2 is intended to work in a similar manner, but in a way that supports the last technology acquired by PT.

Notifications & Custom Mailing

It is intended that VACv2 allow the operator to choose what should trigger a notification, but also to provide control over the text contained in the notifications to be sent.

In VACv2, the operator will be able to choose which action will trigger the notification between starting a scan, ending, or both, while also being able to customize the fields of the notifications between the title, message or both.

Technology Loading

When VAC was developed, it was designed only to handle two scanning technologies as seen before. This requirement is essential because not only PT acquired another scanning technology, but also because VAC got immediately eclipsed.

This requirement aims to not only making VACv2 agnostic to the technology but also giving to it the power to recognize new technologies without needing a significant development or even having to restart the software.

Other Features

Over the next chapters, the reader might be able to notice other features not referenced at this point. This is because the student made additional features over VACv2 with the objective of improving the tool's usability, user-friendliness, amongst others. Some examples of these additional features can be:

- Dashboard which allows one quick visualization of the active scans, and also past scans.
- Development and Production environments.
- Activating a debug mode at any time to enrich logs when trying to find a problem.
- Possibility to observe the appliances statuses.
- Possibility to pause/ resume the appliances.

4.1.2 Adding a new Scanning Technology into VACv2

Among a few other issues, one that immediately stood out was the introduction of a new scanning technology, which VAC was not able to handle. However, after the first objective of this project, VACv2 Improvement of VAC, already make easier the introduction of new scanning technologies into the solution.

Qualys Cloud-Based is the scanning technology to be added into VACv2 and will be described in further detail next. The introduction of this technology will also be useful as a guide for future scanning technologies to be added.

Qualys Cloud-Based

"Delivered as a Public or Private Cloud, Qualys helps businesses streamline their security and compliance solutions and build security into their digital transformation initiatives – for greater agility, better business outcomes, and substantial cost savings." [57]

Qualys is a vulnerability engine technology specialized in providing Security As A Service (SecAAS), it is a sophisticated platform that offers multiple services over the cloud, its software suite is illustrated in figure 4.1. The highlighted modules are the ones who will take part in this master thesis.

Concerning OpenVAS and Nexpose, Qualys being a cloud-based solution in which no effort is required from the DCY's personal, it will relieve them from the usual chores associated with the maintenance of the server.

Next lays some of the advantages which influenced PT when acquiring Qualys' services.

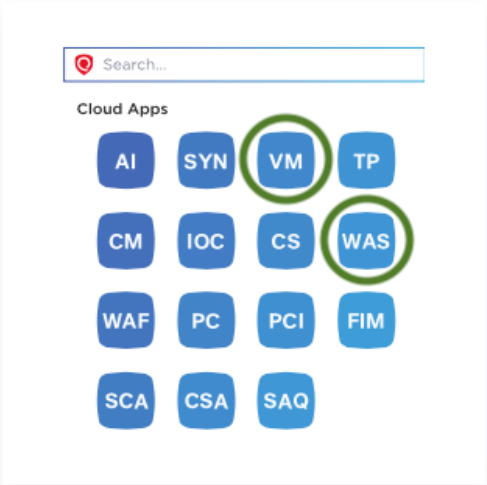


Figure 4.1: Qualys Modules

Cloud-Based This feature is attractive to PT because it allows DCY's personal to disengage from activities like the maintenance of the internal scanner appliances. Besides this feature, it also provides another visibility of the assessments that until this point PT did not have, through the use of external and internal scanning appliances. In this case, the external scanner is the one providing the additional visibility. Another advantage is the service being provided through the cloud, meaning that the operator is able to access the platform wherever he might be and at any given point in time.

Vulnerability Management This module is responsible for performing infrastructure's scans. It works similarly to Nexpose and OpenVAS. However, contains a few differences in relation to the other scanning technologies given its cloud-based character.

Web Application Scanning (WAS) This Qualys' module is responsible for performing scans over web applications. A WAS scan will find vulnerabilities in a website. In this type of scanning, authentication is crucial because much sites provide different views according to the user permissions. WAS is an entirely new scan type to add to VACv2.

VAC was designed to perform only VM scans and to work with in-premises scanners without the concept of having networks assigned to appliances.

VACv2's appliances scanning types will have to contemplate not only what was already mentioned but also will have to associated appliances to networks, in order to function correctly with Qualys. This because PT's internal network is subdivided and Qualys will operate with an appliance by subnetwork plus an external appliance. Meaning that the

method that VAC used to use when assigning a scan to an appliance, which consisted in assigning the scan to the most available appliance, with the insertion of the new scanning technology will not work because Qualys manages its appliances.

4.1.3 Results treating & Data correlation

VAC's Results module was developed in a way which the operator had no control whatsoever over it. At the end of every scan, the results would automatically be uploaded into every DCY's Information Repository. This is an unwanted behavior for VACv2.

VACv2 will have a redesigned Result's module. One that empowers the operator to manage the actual information repositories, while also allowing the operator to choose where the results are to be upload from the available information repositories.

This section represents the last objective of the Continuous Security Assessment project. It is intended that at the end of this objective, VACv2 provides power to the operator for uploading the scans' results into platforms at his choice. It is also intended VACv2's Results Manager module to become unattached from any technologies. In a nutshell, it is expected that this module behaves similarly to the scanners Module. Furthermore, a new technology is to be added beyond the two that already exist, AlienVault USM[13].

However, this objective might be seen as a two-step implementation, because it is supposed that VACv2 uploads the results into Hidra. From the results available in this platform, a data correlation software - Maltego - will be able to fetch them. Maltego will have to suffer a specific development for it to be able to fetch the results from Hidra and correlate the data correctly.

Let's get to know more of these new technologies.

AlienVault USM

AlienVault Unified Security Management (USM) is a SIEM, VAC already handled another SIEM, ArcSight. However, the way VACv2 will establish a connection to AlienVault is entirely different from the way VACv2 work with ArcSight.[2]

Paterva's Maltego

Maltego is a powerful tool that provides a visual layer of data correlated in the form of a graph. This software is highly used by people belonging to the information security sector, and it is used by both criminals as law enforcement agencies. This software was released in 2007 and already makes part of Kali Linux OS software suite.

The initial intent for this project was to connect Maltego and VACv2, in a way that the need for interaction would be reduced to a minimum. Maltego was intended to start the interaction with VACv2 by sending the targets to VACv2. Which would then handle the

[2]The connection between VACv2 and ArcSight was not altered from what was already done by VAC.

scanning process, and after the scan ended, it would provide the results back to Maltego. The reason why this intent was not achieved will be approached ahead.

Maltego can have data inserted manually, or it can fetch data through the use of Transforms or Machines. A Transform is a function which purpose is to make a linkage between the output of such function and the entity which served as input. In other words, a transform will only return data related to the input entity. Image 4.2 illustrates Maltego's environment with the Transforms available for an `IPv4 Address` entity. A machine is no more than a set of Transforms.

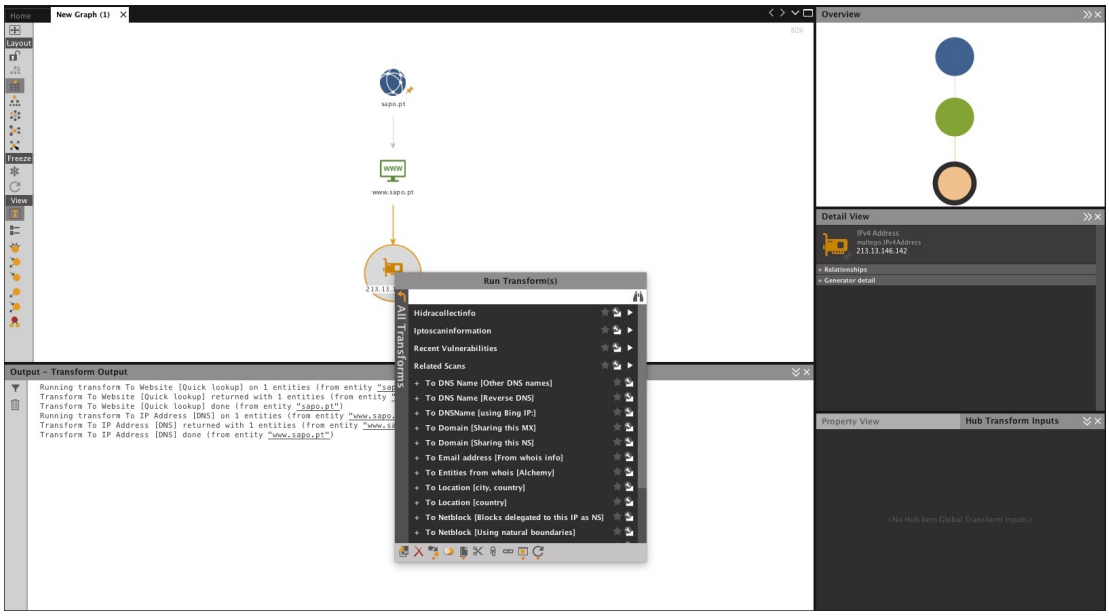


Figure 4.2: Maltego Environment

4.2 Planning

Since day one of the internship until VACv2 entering its development phase, it was almost four months. These four months were spent learning to work with the scanning technologies; VAC; and Maltego.

Learning how these technologies operate played a critical piece in the development of VACv2. The original planning was the following:

- Study and Investigation of the Technologies (8 weeks)
- Initial development version (7 weeks)
- Evaluate scalability and efficiency of the development (1 weeks)
- Identification and Implementation of improvement points (8 weeks)

- Re-evaluation of scalability and effectiveness of the development (8 weeks)
- Elaboration of the Final Report (4 weeks)

The schedule was not successfully followed, and the final solution proved to be somewhat ambitious to a regular program for a master thesis. The next point represents the real calendar as it occurred:

- Study and Investigation of the Technologies (8 weeks)
- Preliminary Report (6 weeks)
- Improvement over VAC structure (5 weeks)
- Evaluation and testing of VACv2 initial structure (1 week)
- Implementation of improvement points (2 weeks)
- Design and development of the Scanning VM module (5 weeks)
- Evaluation and testing of VACv2 Scanning VM structure (1 week)
- Implementation of improvement points (2 weeks)
- Design and development of the Scanning WAS module (5 weeks)
- Evaluation and testing of VACv2 Scanning WAS structure (1 week)
- Implementation of improvement points (2 weeks)
- Design and development of the Report module (5 weeks)
- Evaluation and testing of VACv2 Report structure (1 week)
- Implementation of improvement points (2 weeks)
- Study and Investigation of data to send to Maltego (1 week)
- Design and development of the Report module (3 weeks)
- Evaluation and testing of VACv2 Report structure (1 week)
- Implementation of improvement points (1 weeks)

As it is possible to watch, the original deadlines for the project drifted quite a bit. The reason why this occurred will be explained in the next few paragraphs.

The study and investigation of the scanning technologies, information repositories, Maltego and VAC run without any drifts, but please keep in mind that this step only concerned knowing how to interact with these technologies, and it was not about obtaining an in-depth knowledge of them. It was thanks to this step that was noticed that some of the original objectives had to be altered, specifically the original intent between Maltego and VACv2.

In the beginning, DCY intended to add another module to VACv2, also provided by Qualys, which was the Continuous Monitoring module[3]. However, after some investigation, it was made aware that this module would not work as intended, due to not meeting the expectations of PT. The consequence was the removal of this module from the objectives of this project.

One factor that could also have contributed to the drift of the schedule was the understanding and assimilation of this thesis, which passed into the writing of the preliminary report. It was something that proved to be not so easy given all the technologies that were included, and their objective.

The remaining steps of the plan could be seen as multiple iterations of three steps each, these correspond to the steps two to five of the original planning and represent the biggest drift for this project. It was because of numerous reasons which will be stated ahead.

Ruby and Sinatra

VAC Software was developed in Ruby with resort to Sinatra Framework. Both the programming language and the framework matter because the student did not know anything about them, having to spend more time to learn the language before and along the project development.

VAC Procedural Methodology

VAC used a procedural methodology, and this approach the solution passes by breaking the problem into smaller problems, and making functions which will solve the smaller problem, and so on.

It is the sum of every function that will make the program as a whole. This approach does not make more accessible to learn the program.

[3]This module is focused in providing alerts in real time about modifications in the network. It is available at `https://www.qualys.com/apps/continuous-monitoring/`

Kerberos

Kerberos is offered and handled by an external ruby library. However, the problem lied in integrating it with VAC.

After some unsuccessful attempts, the student found a way to make this feature work, but it was one of the most time-consuming features in the project.

Qualys

Qualys does not offer a ruby library to operate over the API, and no external library was found. The alternative was developing one from scratch to integrate it with VACv2 meeting all the requirements. This custom library was one of the most time-consuming tasks done in the project, but it was also the core of the project. This library only provided access to the modules acquired by PT and necessary to the project.

Qualys and ArchSight

Qualys and ArcSight are both proprietary software but provided by different manufacturers. Despite this fact, ArcSight provides what its vendors named connectors. A connector will establish a connection with another technology, and retrieve useful information back to ArcSight. In this case, the ArcSight connector will connect to Qualys, and download the scan reports available in the scanning platform.

ArcSight offered a connector for Rapid7's Nexpose, but it was concluded that Nexpose's and Qualys' connectors were utterly different. There were two problems with the ArcSight-Qualys connector. The first was because it removes VACv2 from the equation, not allowing post-processing of the data - done by VACv2 - before uploading it into the ArcSight platform. The second reason is that it will upload the results os every scan contained in the Qualys platform, which is also used to perform scans over PT's clients, and therefore not all scan results are wanted in the DCY's information repositories.

Maltego and Python

The original vision of the project was of a self-sustained system, which would update itself continuously. Maltego is a software with a different context from the others handled by this thesis, and so time had to be spent understanding how it operates.

The discovery that Maltego would not be able to work with VACv2 was made at an early stage of the project, while still learning how to operate with the technology. The obstacle was that Mlatego does not offer an API, and is not is not ready to receive requests from other software, as it was intended. The problem would be when VACv2 tried to transmit the results back to Maltego.

Also, when studying the software, it was noticed that Maltego did not offer any support for Ruby, meaning that an alternate programming language had to be chosen. One

supported by the technology was Python, and it was also discovered the existence of a specific Framework - the Canari Framework -, for this programming language, which could be used to develop Maltego's transforms. All these circumstances translate into more time spent learning the language and the framework.

4.3 Conclusion

In this chapter, it was possible to see the objectives that the Continuous Security Assessment project was based. The primary intentions were to improve the original software - VAC -, to extend its capabilities, and handling & diffusing the results through DCY's data repositories. All these objectives would lead to a new version of the software and a new name - VACv2.

The following section described the motives why this project drifted from the original planning. Multiple reasons were provided, but this student believes that the most important ones might have been learning both languages, Ruby and Python.

Despite this, it is also the belief of the student that given the complexity and size of the project, he finds the real-time schedule reasonable.

Chapter 5

VACv2 Design & Implementation

"To competently perform rectifying security service, two critical incident response elements are necessary: information and organization." - Robert E. Davis, IT Auditor.

"Computer security can simply be protecting your equipment and files from disgruntled employees, spies, and anything that goes bump in the night, but there is much more. Computer security helps ensure that your computers, networks, and peripherals work as expected all the time, and that your data is safe in the event of hard disk crash or a power failure resulting from an electrical storm. Computer security also makes sure no damage is done to your data and that no one is able to read it unless you want them to." - Bruce Schneier, Security Expert.

In the last chapter, it was shown the directives of this project, and more specifically of VACv2 development. The restructure would bring back VAC to good use by DCY's operators in the second version of the software. It would allow them to lower their effort in a much labor-intensive assignment, through a significant improvement of the tool, while also making simpler to interact with it. Also, detaching any technologies from VACv2, allowing it to operate with any other technology that might come in the future.

In this chapter, we will get into VACv2's designing and development phases. Please keep in mind that this new solution is based on existing software, so there might be references to how the first software was implemented and works, in order for the reader to understand the work achieved with this thesis.

This chapter is divided into two sections, the first section is focused on VACv2's architecture and its main modules, which will also be analyzed and described. The second section is about the development of the tool, and it is composed of three parts symbolizing the three objectives of this project. Inside each part, there is a description of the development made to each module which got affected by such objective.

5.1 VACv2 Architecture

The VACv2 architecture is based on a crucial circumstance, the difficulty associated with understanding the original VAC solution. The chosen methodology to be applied in VACv2 was the modular programming, this is a technique that tries to divide the program functionalities into independent modules, meaning each module is self-sustain and capable of executing all the tasks that fit into their purpose.

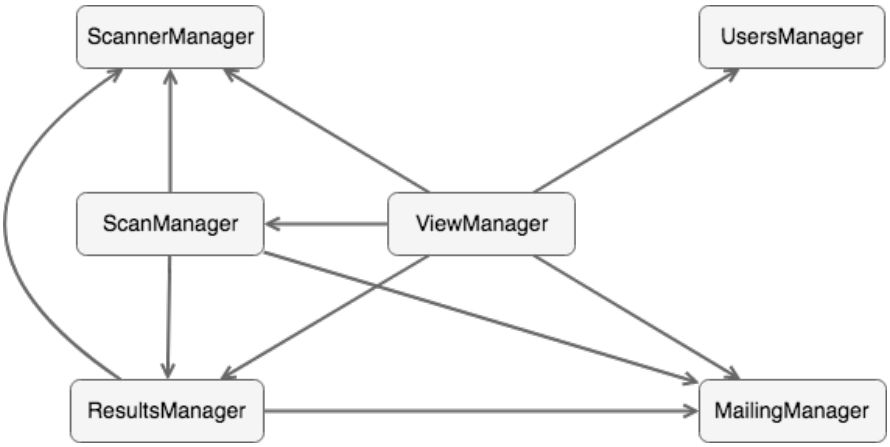


Figure 5.1: VACv2 Main Modules Associations

Image 5.1 exemplifies VACv2's structure. In the next subchapters, every module represented in the illustration will be analyzed ahead, more specifically, it will be described the module's purpose and its associations to other modules.

5.1.1 View Manager Module

This module's responsibilities is to handle every request that comes from the client-side and to reply to such requests.

This module is associated with every other module for natural reasons. If that were not the case, it would not be possible to interact with VACv2, *i.e.*, this object transmits the user's intention into the system. Another of its responsibilities is to launch the web server after a successful initialization of the system. As expected it is not requested by any of the other modules.

5.1.2 Users Manager Module

This module, as the name indicates, is responsible for handling users. Also, is responsible for authenticating the user with resort to Active Directory Kerberos Authentication, and for verifying the authorization of the user to access the solution, in other words, it will authenticate and verify that the user has access to VACv2.

After the authentication and confirming the user's access, it manages the user session while he is accessing VACv2's client-side. The View Manager module is the only one to perform requests to this module. In a first step, it will provide the user information for authentication, and if successful it also provide a session hash for the User Manager to maintain, and at every request done by the user, its session will be verified.

5.1.3 Scanner Manager Module

This module was the backbone of VAC and will remain to be the backbone of VACv2. This module is what enables the communication with the scanning technologies, and without it, there would be no solution since the purpose of the project is to obtain a centralized solution for managing scans over critical targets.

There will be no scans if VACv2 cannot successfully establish a connection to the scanning technology's appliances. This module is accountable for managing the scanner technologies available and their appliances. In VACv2, the user is empowered to configure the scanning technologies, *i.e.*, setting the scanning technologies which are to be considered as for critical use only[1]. This module will also act as a frontier for the scanning technologies, meaning any other module will not be able to make use of the scanning technologies unless if it is required to this module.

There are three modules which request services from this module, which are the View Manager module, the Scan Manager module, and the Results Manager module. The View Manager module is what allows the user to manage this object and its derivatives. Concerning the Scan Manager and the Results Manager modules, these need to interact with this module for being able to launch, track and retrieve information from the scanning appliances about scans or results.

This module makes use of no other module.

5.1.4 Mailing Manager Module

This module came to simplify VACv2's notifications, in other words, this module will allow the possibility of using distribution lists. It is responsible for keeping every distribution list and its content, meaning it will hold to every email associated with a distribution list.

This module is requested by the following modules: View Manager, Scan Manager, and the Results Manager. The View Manager is what allows the management of the distribution lists. The Scan Manager and the Results Manager modules are associated with this module to retrieve the content of the distribution lists to notify about actions regarding those modules.

[1]By critical use it is meant scanning technologies that will only be used to scan the most critical assets, usually because they are paid platforms and provide a higher confidence in the results.

This module makes use of no other module.

5.1.5 Scan Manager Module

If the Scanner Manager module is the backbone of the solution, then this has to be considered the CPU of it. This object is responsible for maintaining every scan configuration, while also being accountable for triggering and managing active scans. Also, for the scans that have integrators configured, this module will trigger the Results Manager module for handling the scan results.

The Scan Manager module requests services from the Scanner Manager, Mailing Manager and the Results Manager modules, and is requested by the View Manager module. For natural reasons, it must make use of the Scanner Manager module, to request scan launches and to track changes to active scans. About the Mailing Manager module, it is only natural that it gets requested about distribution lists when notifications are to be sent regarding scan's actions. The Results Manager module contains the integrators responsible for uploading the scan results into the DCY's information repositories. As the scan configurations contain the property that identifies the integrators which the results are to be upload. At the end of a scan, the Scan Manager module will request the Results Manager module for it to handle the scan results.

The View Manager module makes use of this module, for the same reason as it used all the other modules, to display information regarding the scans to the user, and also to allow the management of scan configurations.

5.1.6 Results Manager Module

From the previous module, we already can infer the purpose of the Results Manager module. This module has more responsibilities than just handling DCY's Information Repositories integrators, it will also parse the scan's results for excepted or excluded events, while also managing every vulnerability event ever reported. This module requests services from the Scanner Manager and the Mailing Manager modules, and is requested by the Scan Manager and the View Manager modules.

This module must make use of the Scanner Manager module to request the generation of a report belonging to a given scan that has occurred, and when the report is generated to download such the report. The Mailing Manager module is requested by this module when notifications are to be sent. These notifications intent to notify the distribution list about the results being available at a given platform.

The View Manager module makes use of this module, for the same reason as all the other modules, to allow the management of the information regarding the integrators. It will also allow the creation, update or removal of vulnerability events to except or exclude. The Scan Manager module requires this module in the scan configuration procedure, and

to trigger the Results Manager module into its normal routine when a scan as finished.

5.2 Implementation

Now that we have seen the architecture of VACv2 - the main modules, and their purpose -, its time we move into the implementation details of VACv2. At this point, what is expected from VACv2 is already known, so in this section, we will see how it was achieved. This section will be divided into three subsections - the objectives of this work -, and at each subsection, it will be provided an explanation of each module that suffered from changes into achieving that particular objective.

5.2.1 Improvement of VAC

The first step of this project was the improvement of the first version of VAC software, to achieve it the solution's coding and its architecture had to be understood.

Following, we will analyze each one of the modules affected by the objective Improvement of VAC. Some of the modules already existed, in these cases, it will be provided a description of what was being achieved by VAC, and after, how the module evolved, and what is being achieved by VACv2.

User Manager Module

As there was no type of user management in VAC, the structure presented of this object was developed to meet VACv2's needs. Illustration 5.2 represents the structure of this module in VACv2. After, we will get into more detail about the classes contained in the illustration.

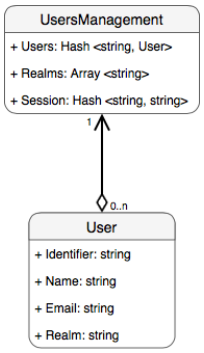


Figure 5.2: VACv2's User Manager module's structure

User This class' purpose is to represent a user that is authorized to access VACv2. The class is composed of four properties, the user id which is the company's login assigned to the user, the name of the user, the email of the user, and the Kerberos REALM the user is associated.

UserManagement This class' purpose is to maintain the information related to the users that may access the solution. In other words, it is responsible for storing the User classes. It will also perform the authentication for the users who try to access the software, and manage the active sessions. This class is composed of three properties.

Users A user's hash, in which the User's id acts as the key and the User class as value.

Realms An array of Kerberos' REALMS. Only the contained ones will be tested for access.

Session A session's hash, in which the key corresponds to the cookie serial and the value is the User's id.

The normal operation of this module is as follows. In an early stage, the user must already have been whitelisted in VACv2, and this is due to the authentication procedure which will require information about the user, *i.e.*, the user login, and the Kerberos REALM.

When the user sets off the login procedure, the User Manager module will try to match the login contained in the provided credentials with a whitelisted user. If it finds a match, then with the help of a ruby gem[2] that handles the Kerberos protocol - named "omniauth-kerberos" -, the user's principal name[3] with the provided password by the user at login will be verified against PT's Active Directory. If the user gets verified, then he will gain access to VACv2.

This module also manages user's sessions. However, it is not responsible for the initial association between the user with the session hash. When the user first accesses VACv2, the View Manager module generates the session hash, and when the user checks out, the session hash will be handed over to this module for it to store. From that point forward, every time the user performs a request his session will be verified.

This module is responsible for making VACv2 achieve three requirements of the improvement of VAC objective, which are:

- Authentication of Users
- Authorized Users (Whitelisting)
- Cookie-based Session - With the synergy of the View Manager module.

Mailing Manager Module

VAC did offer email notifications. However, emails were stated in a non-desirable way. Illustration 5.3 represents the use of notifications in VAC.

[2] A gem is equivalent to libraries in other programming languages.
[3] The Principal Name is the concatenation of the User's id plus the at symbol [@] plus the User's Realm.

Figure 5.3: VAC Email Notification Configuration

From illustration 5.3 it is possible to infer that VAC's email notifications had to be part in every scan configuration, even if using the same emails in multiple scan configurations. The need to perform a single change in the distribution list was a delicate task allied to the fact that VAC did not allow the edition of objects, which means the original object would have to be deleted and a new one created to reflect the change.

The Mailing Manager module is responsible for managing the distribution lists in VACv2, which can be reached out by the other modules. Illustration 5.4 presents the architecture of the Mailing Manager module in VACv2. After, we will examine the structure and purpose of the classes present in the illustration.

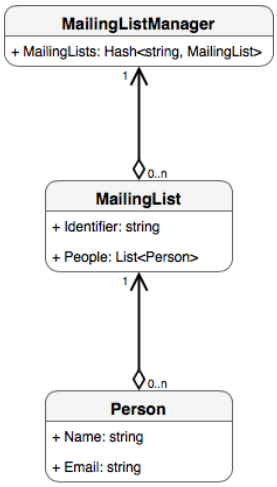


Figure 5.4: Mailing Manager module's internal structure

Person This class is the foundation of this module. Its purpose is the personification of a person to be notified. It is composed of two properties the name of the person and its email address.

MailingList This class aims for the aggregation of people to notify, in other words, it will group Person classes. This class is also composed of two attributes, an identifier, and an array of Person class.

MailingListManager This class's objective is to manage mailing lists. It is composed of a single property, named Mailing list, which is a hash where the Mailing List's

identifier acts as the key, and the Mailing List object is the value.

There is no standard operation for this module, due to its secondary character in the solution. Its purpose is to allow a more user-friendly experience for the operator of VACv2.

The expected behavior of this module is to after the creation of a mailing list, when the user accesses the configuration of a new or an existant scan, to present all the available mailing lists.

This module accomplished the possibility of creating and managing Mailing Lists in VACv2, one of the requirements for the present objective we are analyzing.

Scanner Manager Module

This section will now describe the improvements made over the Scanner Manager module, but before, the reader must know how this module behaved in VAC. VAC was only capable of handling OpenVAS and Nexpose scanning technologies.

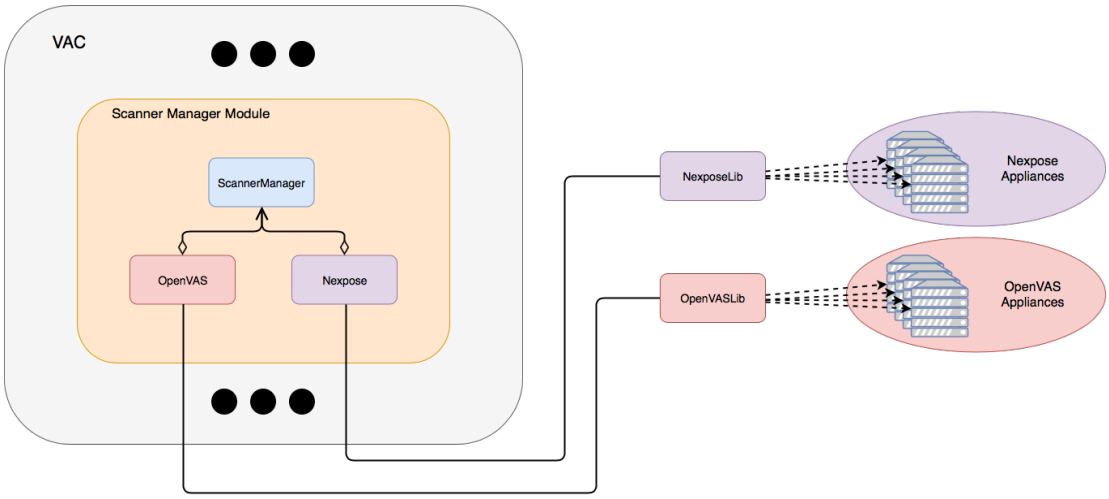


Figure 5.5: VAC's Scanner Manager module internal structure

VAC's Scanner Manager module is going to be analyzed according to illustration 5.5. The ScannerManager class was responsible for storing the information related to every scanner appliance, while also managing them. However, it also kept non-scanner related objects, more concretely scans and reports.

It is possible to observe that the ScannerManager class had a relation of one-to-many regarding OpenVAS and Nexpose classes. Despite these classes having no bound, they were kept in the same property, a hash in which the key was the identifier of the appliance, and the value was the object itself.

OpenVAS and Nexpose classes represented an appliance of that scanning technology, and despite these classes were independent, they were developed to had a similar structure, where both had the following properties:

- Appliance Identifier
- Scanning Technology
- Host
- Username & Password
- Hash - for extra information

It was through these classes that VAC was able to use the scanning technologies appliances, with resort to custom libraries - OpenVASLib and NexposeLib. NexposeLib was the library which provided an API for establishing a communication with the Nexpose appliances, and OpenVASLib represented the same but for OpenVAS appliances.

The improvement of VAC objective required the complete revisit of this structure, and the objective was to empower the user with the capability of managing the scanning technologies from top to bottom. Image 5.6 presents the structure of this module in VACv2, and then we will examine VACv2's Scanner Manager module based on the illustration.

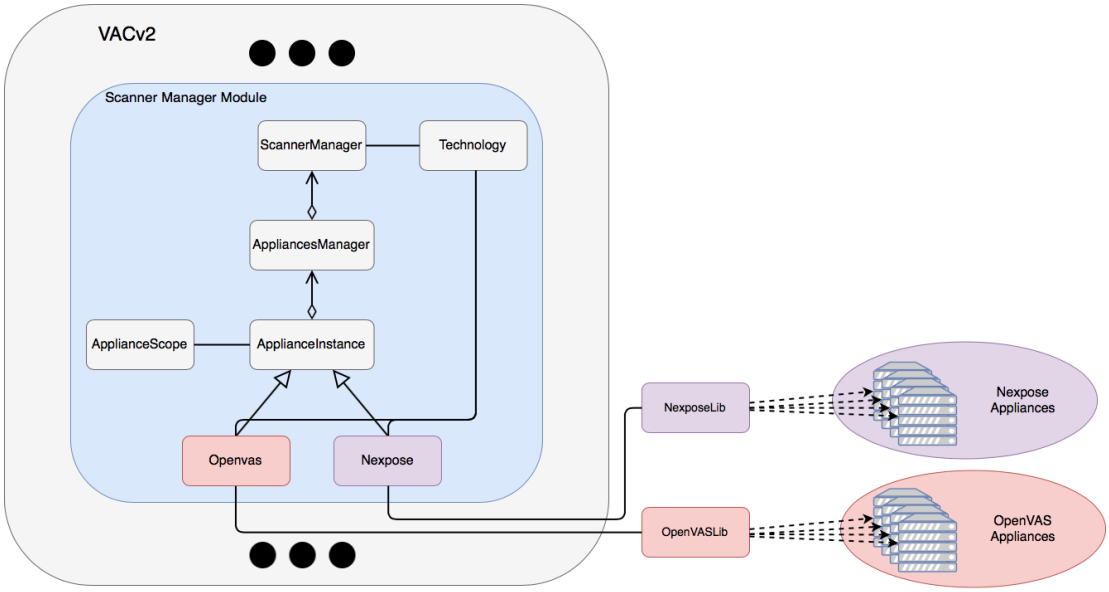


Figure 5.6: VACv2's Scanner Manager module internal structure

Technology The purpose of this class is to allow the system to load scanning technologies classes[4] at runtime, to store them, and to instantiate them when needed. The technology class at system initialization will load every scanning technology class

[4]The term scanning technology class is a reference to the classes that are meant to establish a connection between VACv2 and the scanner technology appliance. These classes can be seen as being a personification of the scanner technology appliances in VACv2.

in conditions[5] of being loaded. After this procedure, the classes discovered become available to VACv2. However, as they were loaded at runtime, they have to be stored by this class and have to be required to this class.

ApplianceScope PT has its internal network divided into multiple virtual LANs (vLAN) to prevent potential attacks from spreading on the entire internal network. A name is provided to every one of the vLANs to identify them more easily. PT's intent is that Qualys contains a scanning appliance in every vLAN.

What this class achieves is the association of an appliance with a network in VACv2. This class is necessary because appliances in different vLANs have different targets for acting. The way this works is through the use of its attributes, which are the following:

Type String with two possible values: "external" or "custom". The "external" value is a reference to an appliance that is outside of PT's perimeter, meaning it will only perform external scans. While the "custom" value is the opposite, meaning it refers to an internal appliance, which can only perform internal scans.

Id This field is set accordingly to the value of the type property. If type value is "external", this field is ignored, but when the type value is "custom", this field will contain the name of the vLAN where the appliance is deployed.

ApplianceInstance Despite VAC's OpenVAS and Nexpose classes having a similar structure, it is possible to acknowledge that it was missing a vital piece, common ground to ensure the same structure between those two classes, and new ones that might be coming.

This class pretends to represent a scanning technology appliance, and it holds the structure that the scanning technology classes[6] will inherit.

The purpose of this class is the standardization of the coding solution. This class will ensure that all scanning technology classes have the same structure. In other words, it will detach VACv2 from any scanning technologies classes, this way not needing any development to VACv2's core to support new scanning technology classes. This class is composed of the following attributes:

- Appliance id
- Host

[5]For a scanning technology class to be loaded, there are a couple of requirements that need to be met, or the Technology class will not recognize it. These requirements are: At a specific directory, it needs to exist a subdirectory which name has to be equal to the class that will be loaded, and this class has to be placed inside this subdirectory.

[6]These scanning technologies class are the ones discovered by the Technology class.

- Appliance Scope
- Username & Password
- Hash - for extra configurations

OpenVAS & Nexpose Each of these classes, will handle its scanning technology for VACv2. These are the classes that will be recognized by the Technology class, which means that these are also the classes that will have to extend the ApplianceInstance class, meaning that OpenVAS and Nexpose classes will inherit the methods belonging to the ApplianceInstance class.

These classes will then handle the inherited methods according to its scanning technology demands. In a nutshell, these classes will receive the input from VACv2 and translate it to the scanning technologies through the custom libraries.

AppliancesManager This class's purpose is to provide control over a scanning technology to the operator of VACv2, while also to be a technology aggregator, meaning it will exist one AppliancesManager object by Scanning Technology.

One responsibility of this class is to store every information relative to the technology. Another responsibility is to store and manage every ApplianceInstance object concerning the associated technology.

The ApplianceManager class is composed of the following properties:

- Scanning Technology
- Modules' List - List of the available scanning types of the technology. In other words, the possible scanning types to perform. *E.g.*, Vulnerability Management.
- Critical - Flag to indicate if the technology is to be used for critical scans only.
- ApplianceInstance's Hash
- Subnetworks' Array - Subnetworks where the available ApplianceInstances objects in VACv2 belonging to this scanning technology are operating
- Templates' List - Scanning templates available to this technology
- Hash - for extra configurations

Scanner Manager In VACv2, the context of this class was reduced to scanner-related functions and objects, meaning it was liberated from every other object not strictly related to scanners, *i.e.*, scans or reports. In other words, the scanner manager will no longer be responsible for supervising active scans, or reports. This class's purpose is to act as VACv2's interface for the scanners structure. This object is the point of entry of any request made to the Scanner Manager structure.

Concerning the old procedures of supervising active scan or reports, in VACv2 this object will be requested by the responsible modules to perform operations or provide updates related to their tasks in the scanning technology's appliances.

This class is composed of two properties:

- ApplianceManager's Hash
- Hash - for extra configurations

The normal operation of this module is as follows. At software initialization, the Technology class will find the available ApplianceInstance specialization classes, load and store them. The ScannerManager class is associated with the Technology class, and this association is what allows for the instantiation of the classes that extend the ApplianceInstance.[7] The next step is loading the previous state of the software, which will only occur when the system detects the existence of files used for saving the previous state.

After system initialization, the ApplianceManager class periodically make requests to each appliance for monitoring purposes, *e.g.*, for testing the connection to the appliance, or for knowledge of the number of scans currently active, *et cetera*. Three entities interact with the Scanner Manager module:

- Scan Manager module
- Results Manager module
- View Manager module

The Scan Manager makes requests to this module for launching new scans, and to get information about active scans. The Result Manager makes requests to this module for launching new reports over recently finished scans, for tracking the progress of the report generation and when it finishes, to request the download of the report. When the report is downloaded, the class that supersedes the ApplianceInstance will handle the raw data provided by the scanner technology into a VACv2's internal object. If this were not the case, VACv2 would still be attached to the scanning technologies. The internal object is named ReportInstance and will be approach ahead.

The View Manager enables the interaction of the user with the module, but also is responsible for improving the user experience in the client-side, *e.g.*, when configuring a scan, the user will have at its disposal a checkbox related to the scanning technology that will affect the remaining properties if enabled or disabled. The previous example tries to transmit that all the fields displayed in the client-side of the solution will be previously loaded, to make the user make the best decision with the smallest effort possible. The

[7] As the specialization classes are not loaded at initialization phase but rather at running phase, they cannot be instantiated as the other classes because they are not available to the system.

View Manager will allow the user to customize all the configurations from the Scanner Manager to the OpenVAS or the Nexpose. Also, will be able to create, update or remove ApplianceInstance's objects, with an additional feature of pausing or resuming, the interaction of VACv2 with the correspondent appliance.

Another situation is the fact that this module is now able to match a target to a network. When a user is configuring a new scan, and after the information is submitted, before being accepted, the Scanner Manager module will validate the targets. The validation consists of verifying if the targets are valid for scanning, it will try to match the targets with a network - internal or not. The targets can be scattered over different networks. If a target is set in a network that VACv2 contains no appliance to operate, or if it is contained in a signaled range that is not for scanning, then the target will be marked as non-valid, and therefore the scan configuration will not be accepted.

In VACv2, the operator can choose what scanning technology will perform the scan by tunning two properties while making the scan configuration, which are:

Critical A check property. When is checked, it means that the scanning technology to be used is one of the marked for critical assets scanning.

Templates This property is affected by the previous one, it will only present the templates belonging to the scanning technologies which are flagged according to the critical property. When the critical property is checked, this property will only present templates from scanning technologies marked in VACv2 as being for critical use only. If the critical property is not checked, then it will present the templates belonging to the technologies that in VACv2 are not marked as being for critical use. When choosing a template, they are separated by technology, allowing the operator to know which technology will perform the scan. Figure 5.7 illustrates this property as it is presented in the client-side.

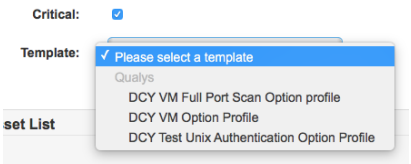


Figure 5.7: Critical Scanning Technologies' Templates

The improvement made to this module increased the functionality of the software in quite a bit. Concerning the first objective of this project - Improvement of VAC-, this module has achieved the following requirements:

- Automatical Identification of Targets
- Loading Technologies at runtime

Scan Manager Module

This particular module did not exist in VAC. Scans used to be managed with resort to two other modules: the Scheduler and the Scanner Manager modules. Each one of them had distinct tasks delegated.

The Scheduler module was responsible for storing every scan configuration, and also to trigger the launch of scans when the time came. The action of launching scans consisted in passing the scan configuration into the Scanner Manager module, which marked the end of the association of that scan with the Scheduler module.

From that point on, the Scanner Manager module would become responsible for launching the scans, while keeping track of the active scans. When the scans ended, the Scanner Manager module would request the generation of the correspondent report, download it when available, parsed the results and then delivered the data to the Results Manager module.

VACv2's Scan Manager module contains the focus of only being concerned with scan-related objects. This module can be seen as an evolution of VAC's Scheduler module. However, let's start by analyzing VAC's Scheduler module structure illustrated in figure 5.8.

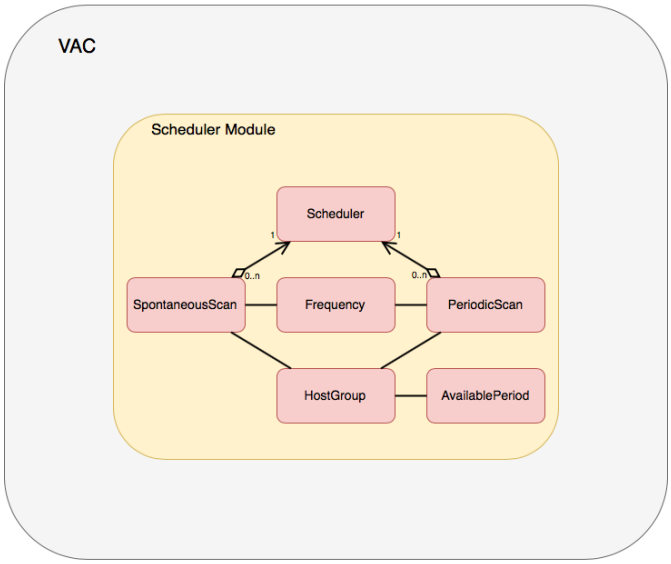


Figure 5.8: VAC's Scheduler module internal structure

AvailablePeriod This class intended to achieve a concept that exists in some scanning technologies. This class established an association between a host group with an availability period to perform the scan. In other words, the scan could only occur inside a specific time frame. If the scan exceeded that frame, then it would be paused, and when back inside the window, resumed.

HostGroup This class's purpose was to aggregate the hosts that were the targets of the scan. In VAC, it was only possible to make VM scans, which means that the target could only be IP addresses.

Frequency This class allowed to set a frequency to the scan. It was also used for determining the next time for the scan to occur after the present iteration. However, the calculus done was somewhat rudimentary, it would just sum the number of days corresponding to the frequency option chosen.

SpontaneousScan & PeriodicScan Both these classes try to personify the same, a scan configuration. Their purpose was to gather all the information regarding a scan. Besides the associations with the Frequency and the Hostgroup classes, these classes are composed of the following properties:

- Scan Name
- Scan Template
- Hash - for extra configurations, *e.g.*, the criticality property.

Scheduler The Scheduler class, contained two purposes. The first was to store all the scan configurations - SpontaneousScan and PeriodicScan classes. The second was to launch the scans when it reached their scheduled time. The launch of the scan consisted in passing the scan configuration to the Scanner Manager module while also requesting this module to launch the scan over that configuration, thus finishing its interaction concerning this scan.

In VAC the lack of a specific manager for handling scans was considered to be urging, given that in VACv2 the scanning type scope was to be augmented, leading to the development of the Scan Manager module in VACv2. VACv2's Scan Manager module is illustrated in image 5.9.

As the reader might notice, VACv2's Scan Manager module structure is significantly different from the structure VAC used to use to handle scans. At a glance the differences are:

- Absence of the Scheduler class
- Removal of the PeriodicScan and SpontaneousScan classes
- Replacement of the HostGroup class
- Removal of the AvailablePeriodic class

Let's now analyze this changes in more detail. The Scheduler class got directly replaced by the ScanManager class. The PeriodicScan and the SpontaneousScan had the

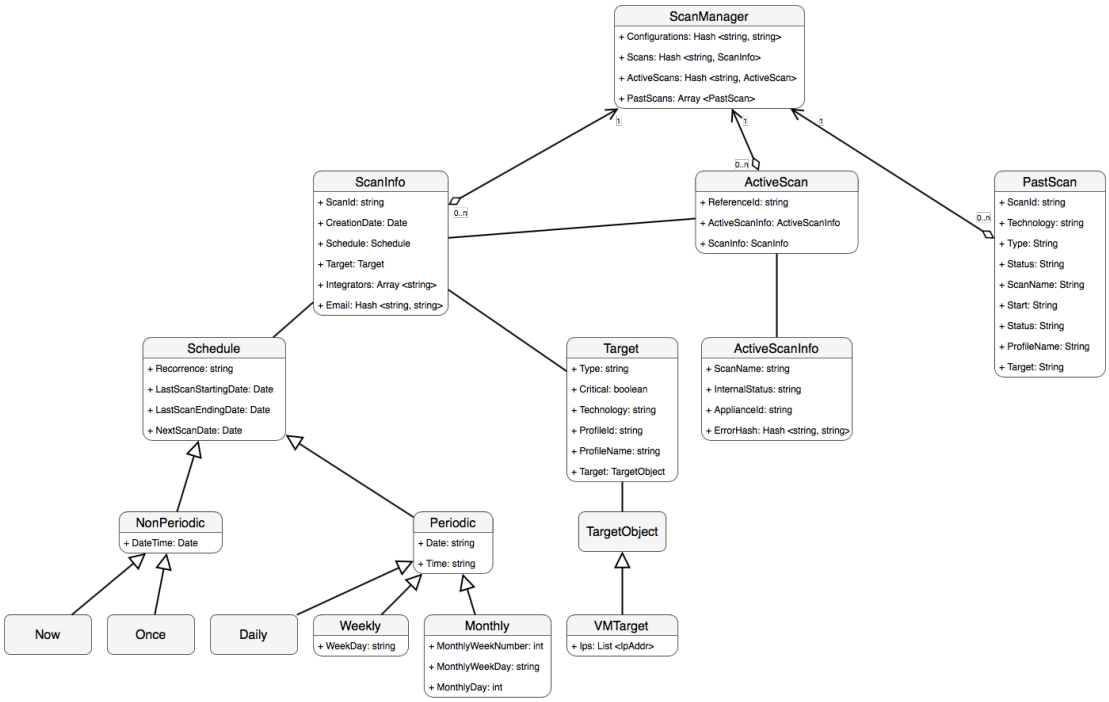


Figure 5.9: VACv2's Scan Manager module internal structure

same purpose with a different recurrence, so these two got replaced by the ScanInfo class, which contains an association with the Schedule class for setting the scan recurrence. The HostGroup class got replaced by the Target class, and this class contains an association with a TargetObject, which its purpose is to allow multiple scanning types in VACv2. Finally, the AvailablePeriod was entirely removed from the solution, not because of the feature itself, but because DCY's operators did not use that feature and requested its removal.

The classes illustrated in figure 5.9 are described in more detail next.

ScanInfo The scan recurrence was no motive for handling scans in a different manner, meaning that VACv2's ScanInfo replaced VAC's SpontaneousScan and PeriodicScan classes.

To allow multiple scanning types and ensure that the ScanInfo class does not get attached to any scan type, the ScanInfo will aggregate all the scan data. Its properties are:

- ScanId - This property holds the Scan Identifier in VACv2.
- CreationDate - This property will contain the date and time of when the Scan Configuration was created.
- Schedule - This is a pointer to an object that will contain all the information related to the frequency and scheduling of this scan configuration. It will be analyzed ahead.

- Target - This is a pointer to an object that will contain all the information related to the type of the scan and the target of the scan. This object is the key to the ScanInfo class getting detached from any scan types, this way making the Scan Manager module more scalable. This object will be analyzed ahead.
- Integrators - This is an array that will contain Integrators' identifiers. It will point to the integrators to use when uploading the scan's results. However, this property is related to the third objective of this project, meaning it will be analyzed ahead.
- Email - This is a hash where all the notifications-related data is stored. From the MailingList's Id to custom titles or custom messages. Also in VACv2, the Operator is allowed to choose which scan states will trigger a notification, this information will also be stored in this hash.

Schedule This class's purpose is to associate a frequency type with a scan. It keeps all data related to the scheduling of the scans. It is composed of four properties:

- Recurrence - Contains more of an informational character, holds the type of schedule, its possible values are: Now, Once, Daily, Weekly, or Monthly.
- LastScanStartingDate - Last time that a scan was launch based on the configuration available in the ScanInfo.
- LastScanEndingDate - Ending time of the scan belonging to the LastScanStartingDate property.
- NextScanDate - Future date to launch the associated scan configuration.

NonPeriodic & Periodic Both these classes extend the Schedule class, each one of them holds to properties suitable to its frequency type.

For a non-periodic scan, all that is needed is a DateTime property. This property's type is internal to Ruby and will hold the date and time in the same property, its value will correspond to when the scan configuration associated should be launched.

For a periodic scan, there is the need to store both the date and time. However, in this case, the date and time cannot be stored in a DateTime property, being stored in two separate objects. The time property contains the purpose of indicating the time at which the scan should be launched. However, the date property will not be that straightforward. The date property will act as a restraint to the scan, in other words, the scan will only be unblocked when the date value has passed.

Now, Once, Daily, Weekly, Monthly These classes are an extent to the types of schedules that exist in VACv2, the Now and Once classes extend the NonPeriodic class, while the remaining extend the Periodic class.

The Now, Once and Daily classes only contain the properties inherited by their superclass, the Weekly and Monthly classes have more properties given their extra complexity. Let's now analyze each case:

Now & Once The DateTime property will hold the schedule of when to launch the scan configuration. The difference between these classes is that the Now class will automatically fulfill the property value with the present time, while the Once class will let the operator choose the date and time.

Daily The date property will restrain the scan has already said, blocking the scan from being launch until that date is reached, in that day and each day forward, a scan will be launched at the value contained in the time property.

Weekly The date and time properties work the same as the Daily class, but this class contains one more property, the weekDay. This property will contain the day of the week the scan is to be launched and will launch the scan every week only on that weekday.

Monthly The most complex structure of all. The date and time properties work the same as the Daily class. However, this class contains three more properties because it contemplates two cases. The first situation is about working with the week number and weekday, while the second will work over the day of the month, the next example will help to understand how it works. *E.g.*, every month third Tuesday, or every month fifth.

Each of the Periodic classes will compute the next date for the scan to occur after the previous scan finishes and will do it accordingly to the class's properties.

Target The purpose of this class is to detach the ScanInfo class from being scan type specific. This class will collect the properties which are common to every scanning technology, the properties are:

- Critical and Technology - Allows to uniquely identify the scanning technology that is to perform the scan.
- Type - The scan type to perform.
- ProfileId and ProfileName - The scan template to be used.
- Target - A TargetObject object, this is the object which will contain the scan-specific properties to be used in the scan. It will be analyzed ahead.

TargetObject To detach the Target class from having to know every scan type that might come to exist, which means to prevent the addition of every property required to each scan type, a couple of measures had to be made. The first step was the creation of a class to hold to those specific scan-type properties. However, when a new scan

type was to be added into VACv2, that class would need an additional development, which is not desirable. Another way is to make a class to be extended by more specific classes, this way the Target class and this class do not have to be altered, ever.

The purpose of this class is to make a "bridge" between the Target class and the scan-specific classes, and as there are no common properties between scan-type classes this class will contain no properties.

VMTarget This class can be seen as a replacement for VAC's HostGroup. It is pretended that this class collects all the necessary information related to a Vulnerability Management scan.

VM scans only need the IP addresses that will be the target of the scan, and then this class contains a single property, an array with all the IP addresses that are the target of the scan.

ActiveScan As previously seen, the ScanInfo object is responsible for containing all the scan data. However, when launching a scan, other properties need to be taken into consideration. Also, the ScanInfo object will continue to have data that matter while the scan is running and after the scan finishes, this means that the user cannot change this object when the scan is launched to prevent inconsistencies.

The ActiveScan class contains a similar purpose to the ScanInfo class in the way that it will aggregate all the information related to a scan instance, which in other words, means that this class will represent an active scan instance in VACv2. This class will contain every information required by VACv2 for it to be able to manage every step from the launch of the scan until the result's processing. This class is composed of three properties:

- ReferenceId - In any scanning technology when a scan is launched, it will be assigned an identifier to the scan commonly known as referenceId, this property will hold to that value. In other words, this property will identify the active scan instance in the scanning platform.
- ActiveScanInfo - This is a pointer to an object that will contain all the information related to the scan instance. This object will be analyzed ahead.
- ScanInfo - At the creation of the ActiveScan class to prevent any inconsistencies, as mentioned at the beginning of the description of this class, a clone of the original ScanInfo object is made. This ScanInfo class is a pointer to the clone object. This way, if changes are made when the scan has been initiated, it will only be reflected in the next scan.

ActiveScanInfo This class's purpose is to store VACv2's internal information related to the active scan instance. This class is composed of four properties:

- ScanName - When a scan is launched, the name assigned to the scan will consist in a concatenation of static prefix - "VAC-SCAN" - plus a suffix which corresponds to the date and time of the scan. This property is equivalent to the ActiveScan's referenceId, but this is more user-friendly, and will uniquely identify the scan instance.
- InternalStatus - Each scanning technology contains its scanning states. The specific class which extends the Scanner Manager module's ApplianceInstance class will make a mapping between the technology's scan's states and an internal state belonging to VACv2. Its values range from Scheduled to Running and to Complete in case of success, or Error in case of error. This property will contain VACv2's scanning state.
- ApplianceId - When launching a scan, there will be an appliance that will be in charge of the scan, this property will store the VACv2's ApplianceInstance id.
- ErrorHash - When a scan fails, usually some other information is provided regarding the motive why the scan as failed. This hash will store the information provided by the scanning technology, as well as it will store any errors that might occur while processing in VACv2. The information stored in this property will be presented to the operator.

PastScan This class intends to provide information about past scans to the user. All its values were already mentioned in previous classes. The difference is that all the properties contained in this class are inherited from the ActiveScan's properties, and these will be changed into strings, given the informational character of this class.

ScanManager This class can be seen as an improvement over VAC's Scheduler class. This class is equivalent to the ScannerManager class in the Scanner Manager module way that it is the interface class that will handle any requests that might come to this module.

The ScanManager class contains the purpose of storing any scan-related object, to manage active scans - from the launch until the finish -, and to trigger the Results Manager module to handle the scan's results if the scan is successful. This class will also trigger notifications for the scans which have notification active.

This class is composed of four attributes, and all have been analyzed except for the configurations' hash, this property is for setting up some extra configuration that might come in the future.

The normal functioning of this module is explained over the next few paragraphs. The ScanManager class will periodically perform a routine check to verify if any scan configuration needs to be launched, or if any active scan is in a requiring attention state.

If at least one scan configuration has reached its launching time, a new ActiveScan object will be created, and it is requested to the Scanner Manager module to launch the scan. Concerning the ActiveScan objects, the requiring attention state means that the scan has finished. When a scan finishes, it means that, or the scan is complete or the scan stopped for some reason, in any way, only in case of success the ActiveScan object will be passed to the Results Manager module. Finally, and regardless of what the final state of the scan is, a PastScan object will always be created based on the ActiveScan object.

The explanation of the module made it easy to understand when the Scan Manager module performs requests to the Scanner Manager and the Results Manager modules. The Mailing Manager module is used at the launch or the end of a scan - if such was configured. The Mailing Manager module is required to retrieve the distribution list's emails where the notifications are to be sent.

To avoid the possibility of getting the solution into an inconsistent state, there is no other way to configure new scan configurations but through the client-side.

Considering that these changes were provided in the context of the first objective of this report - Improvement of VAC. This implementation achieved the following goals:

- Correlation Data Integrators[8]
- Improvement of Scan Recurrence
- Notifications & Custom Mailing

View Manager Module

The View Manager module did not exist in VAC. This module is responsible for connecting every module to the presentation layer, while also firing up the web server responsible for handling the clients' requests. Image 5.10 will illustrate the structure of this module.

The View Manager module contains a pretty simple architecture being a single class - the ViewManager class - associated with all other modules. Otherwise, the exchange of information between the client-side and the server-side would not work. The ViewManager class is also associated with the web server.

The ViewManager class will unite VACv2's modules with the VACv2App web server and fire it up. The VACv2App is an extension of "thin"[20], a fast and simple web server developed in Ruby. The web server will set up a few configurations to ensure a more secure communication with the client. Configurations like setting the certificate and key

[8]What was accomplished at this point, will be mentioned in the third objective of this project.

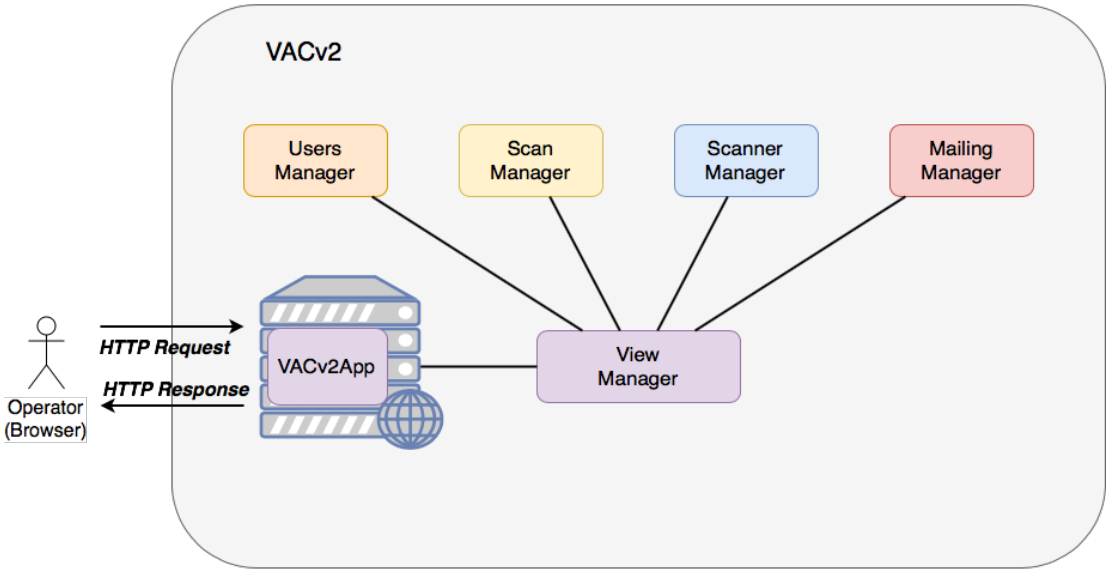


Figure 5.10: VACv2's View Manager module internal structure

to be used in the communication with the client, or setting the strict-transport-security header or the secure flag in the cookies, are a few examples.

The web server is responsible for handling the client's requests. It will receive the requests, validate them in a first instance, and then pass these requests to the responsible VACv2's modules. When the module finishes executing, it will get back to the VACv2App with the answer to be provided to the client, and VACv2App will build a response object an send it to the client.

The VACv2App web server is also responsible for assigning cookies to the users, this is done with resort to a Ruby library name "SessionAuth". When the client authenticates successfully, the cookie is stored by the UsersManager module.

In the end, this module contributed to the achievement of the following goals:

- Cookie-based Session
- Data Pre-Load
- Improvement of Data Management (Creation, Update, and Removal)

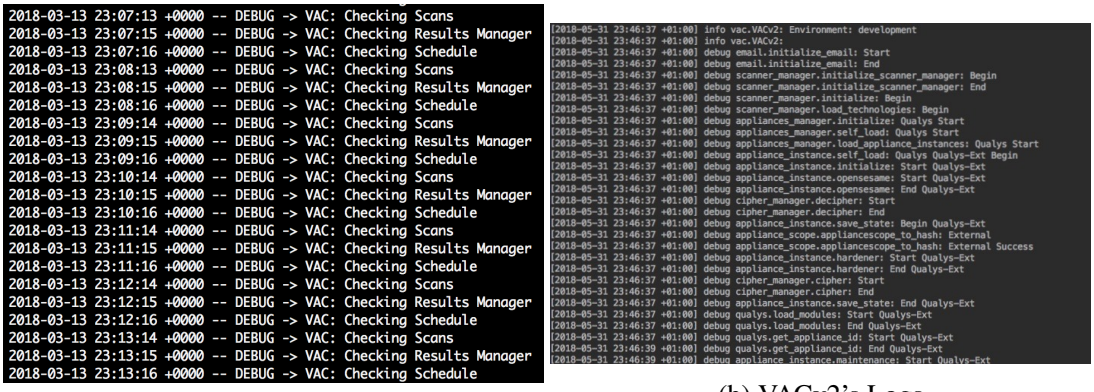
Other Modules

Besides the modules already approach, through the development of this objective, some other modules were made to achieve specific purposes. Next, there will be a small description of these extra modules, and what it was possible to obtain with their assistance.

Logging Module This module as the name indicates is related to the logging of VACv2 software. This module is scattered over the entire solution. Its logs are intended to help in the analysis of a problem or to find a potential bug. One of the goals at this stage of the project was precisely to improve the logging of the solution.

Logging in VAC consisted in creating an object at initialization phase, and that object would be passed at every method call as a parameter. In VACv2 the logging module is a singleton, which means it is available to any object in the solution.

Illustration 5.11a is showing an example of VAC's logging, while image 5.11b is presenting a piece of logging supplied by VACv2.



(a) VAC's Logs

(b) VACv2's Logs

Figure 5.11: Scanner Manager Module Interactions

Ciphering Module The ciphering module did not exist in the original VAC solution, and its purpose is to cipher VACv2's critical information, which meets directly one of the goals for this objective of improvement of VAC. This class is a singleton, and each class that contains sensitive information must make use off this module, the way it works is when a class is saving its state into a file, and one of its properties includes sensitive data, *e.g.*, a password, it would call this module to cipher the password. On the reverse, when loading one of the classes that contain sensitive information, then it would use this module to decipher the password.

Email Dispatcher The email dispatcher module did not exist in VAC, the Scheduler class was in charge of building and sending the notification emails. In VACv2, this class, which is a singleton class, provides the email functionality for all the solution.

Three modules are using the increased functionality achieved with this class in VACv2, which are:

Logging module When a critical log is written, the Logging module will also send an email notifying the owner of the solution.

Scan Manager It is the class that triggers the start and end of a scan.

Results Manager Notify that the scan results were uploaded to a platform and are available. However, we will get back to this ahead.

Network Manager This singleton class is only used by the Scanner Manager module's ApplianceScope class, for consulting which networks are associated with an appliance.

At startup time it will load a file with the network structure into memory, the file is illustrated in image 5.12. In the file, a set of ranges are assigned to an internal network, identifiable by the network name, and so on.

Figure 5.12: VACv2's Network Manager boot file structure

When requested by an ApplianceScope class to find if any of the ranges include the address passed as an argument, it will look through all the ranges, and if the IP address is in range of any of these networks, it will then provide the name of the network.

Client-Side

The frontend side of the solution had to be changed, to reflect the new functionalities and improvements made over the solution. This section will briefly describe the client-side solution and what is now available to the client on each page. Please keep in mind that the next illustrations already reflect the final state of the project. However, only the changes that were originated on the actual objective - Improvement of VAC -, will be addressed.

The original client-side solution used Bootstrap 3[2], which is a framework for developing HTML, CSS, and Javascript. Each of the previous programming languages serves a purpose. The first is for building the web pages, the second for styling the web pages, and the last for programming the web pages. The client-side solution of VACv2 kept the same framework.

Let's now start analyzing the differences in the client-side solution.

Login Page

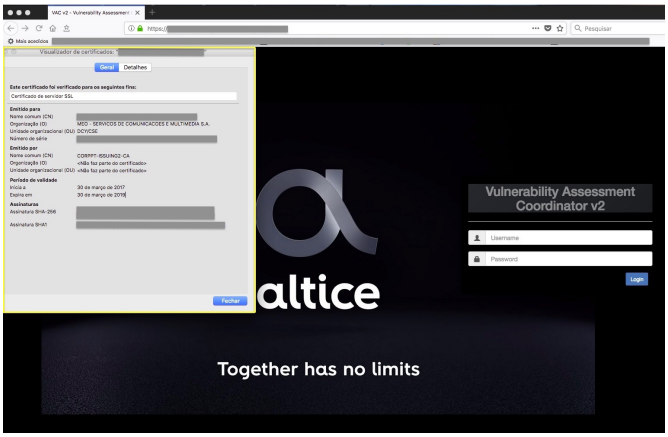


Figure 5.13: Login Page

Illustration 5.13 corresponds to the new first page when accessing VACv2. The highlighted popout is illustrating that server-client communication is ciphered.

Dashboard Page

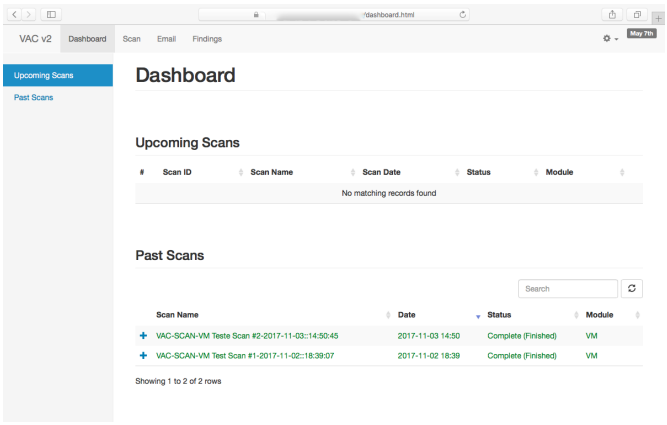


Figure 5.14: Dashboard Page

Illustration 5.14 is the result of the redirect action after successful login into the VACv2. In this page, it is displayed two tables. The first table presents the scheduled scans for the day and their current state, while the second table presents the past scans that have occurred.

Illustration 5.15 is exemplifying the type of data accessible from the upcoming scans table. It not only presents the scans scheduled for the day as it will track down their current state.

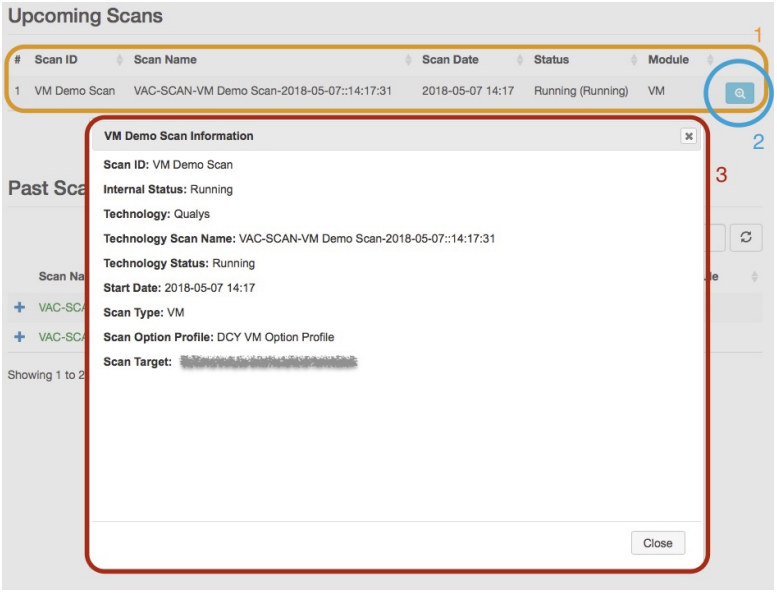


Figure 5.15: Dashboard's Upcoming Scans' Table - Scan Detail

Identified by number one is the table where the scheduled and active scans are, and provides enough information to quickly allow the operator to identify if something might have gone wrong or not. The button illustrated by number two will present more information regarding the associated scan, as exemplified by the popout identified by the number three. The popout provides a more detailed view of the scan, allowing the operator to see the scan configuration, while also presenting information that will allow him to track the scan in the scanning technology as well.

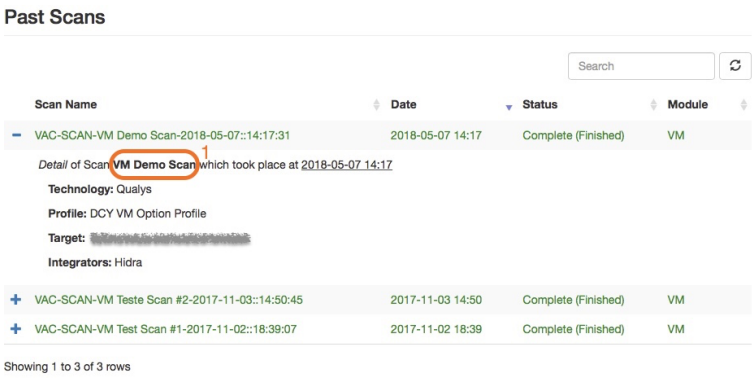


Figure 5.16: Dashboard's Past Scans' Table - Detailed View

Illustration 5.16 reflects the bottom table which is the Past Scans table. The purpose of this table is to provide a historical list of every executed scan. The reader might even notice that the rows displayed are in green, this is an indicator that scan ran successfully if it was presented in red, then the scan would have finished with an error state.

From this table, the operator will be able to tell which was the scanning technology in charge of the scan, when the scan took place, the targets of the scan, where the results have been uploaded to, and so on. The highlight identified by number one pretends to show the reader the Scan Id of the scan configuration, which is the identifier of the scan configuration in VACv2, and the scan name illustrates a scan instance that has occurred for that scan configuration.

Administration Menu

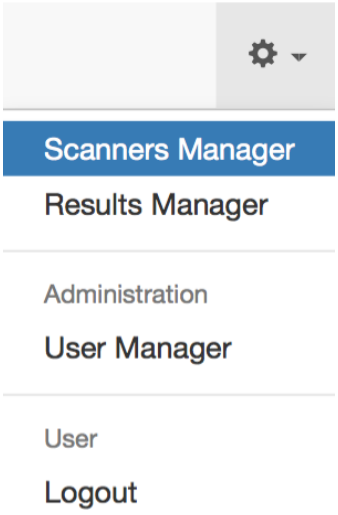


Figure 5.17: Administration Menu

The Administration Menu - illustrated in figure 5.17 - is available through the cogwheel placed in the upper-right corner of each page. The pages available at this place are more related to the management of VACv2 rather than the tool operation.

User's Page

Illustration 5.18 presents the page that makes possible the management of users. There are two tables available, the first is related to the users, and the second related to realms.

For a user to access VACv2, it must be included in the user's table. The Realms' table ensures that VACv2 will only try to authenticate users located in those Realms. If a user contains an association to a Realm that is not specified in this table, then he will not be able to login into the app.

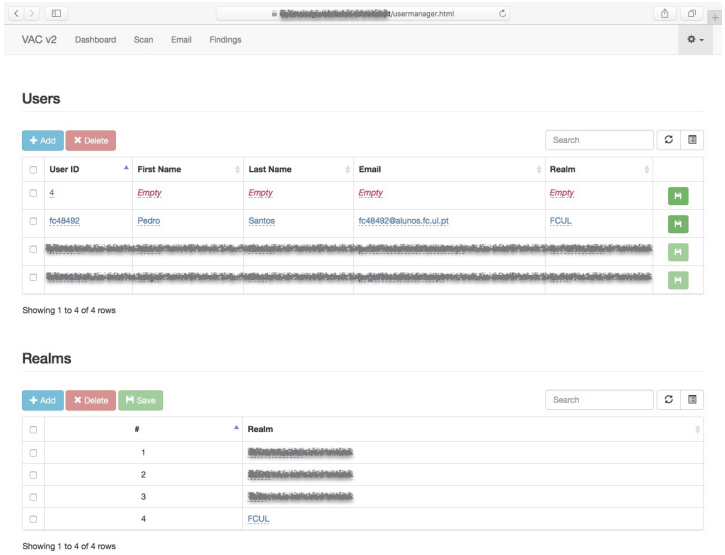


Figure 5.18: Users' Management

Scanner's Page

Illustration 5.19 is related to the management of the scanning technologies and its appliances. The top table portrays the scanning technologies which were recognized and are now available to use, while the second table portrays the scanning technology's appliances added into the system.

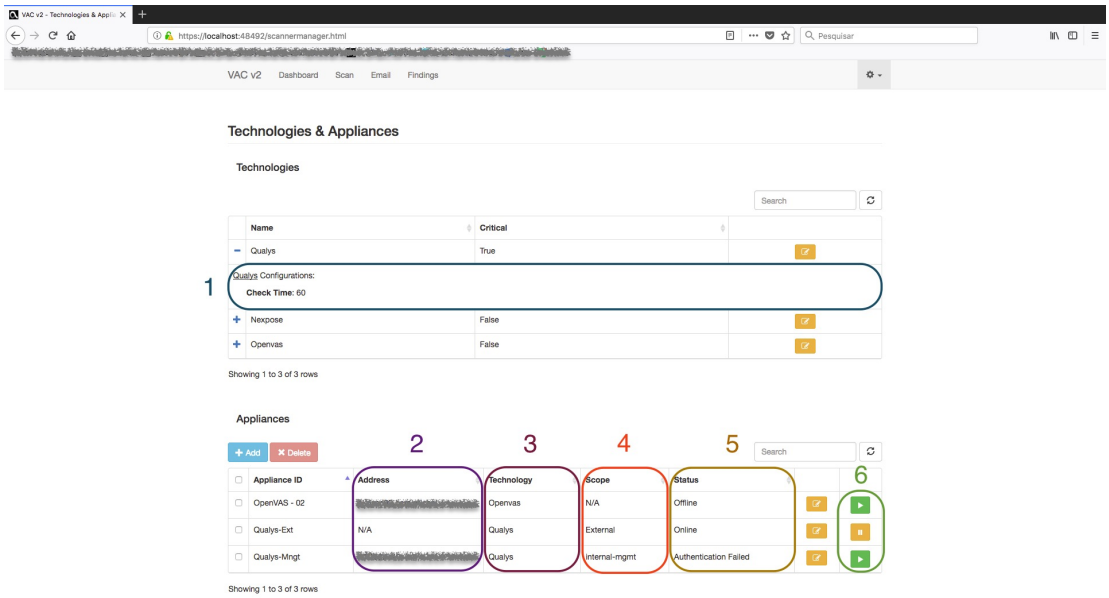


Figure 5.19: Scanners' Management

From the Technologies table is possible to understand which technologies are being used for critical scans, also identified by number 1 is the extended view of the table which shows the additional configurations associated with the scanning technology.

The Appliance's table quickly depicts the overall situation of the existing appliances. Identified by the numbers two to six is presented information related to the appliance and its status. Let's address the meaning of each highlighted column:

Number Two IP address of the appliance.

Number Three Appliance's technology.

Number Four Appliance's scope, in other words, refers to which network the appliance is deployed and is responsible.

Number Five Status of the connection between VACv2 and the appliance.

Number Six Pause/Resume system. If the appliance is paused, then VACv2 will not communicate with this appliance.

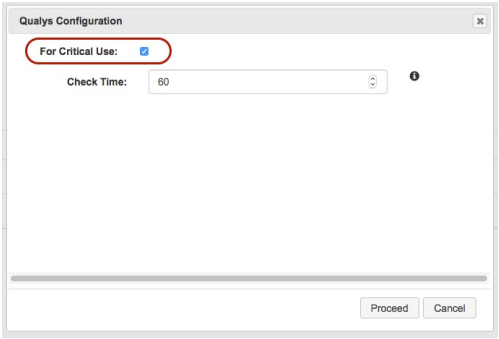


Figure 5.20: Technologies' Table - Technology Detail

Image 5.20 illustrates the update panel of a technology. It is not possible to add new configurations over the interface only to edit the current ones. Additional configurations can only be added in the solution's back-end files, and this is because these configurations, most likely, will have to be referenced in the solution's coding. Highlighted in red, is the checkbox that will activate or deactivate the technology for being used in critical scans. VAC had this feature hardcoded into the solution.

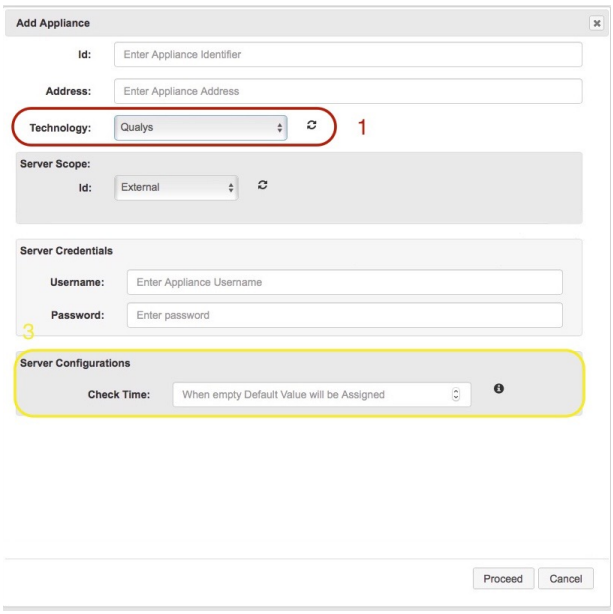


Figure 5.21: Appliances' Table - Add Appliance

Image 5.21 illustrates the addition of a new appliance. The procedure is pretty straightforward, and only the elements highlighted might need extra explanation about what their purpose is. When choosing the Technology value, the Server Configurations section will sync automatically according to the additional configurations configured for that scanning technology.

Mailing List Page

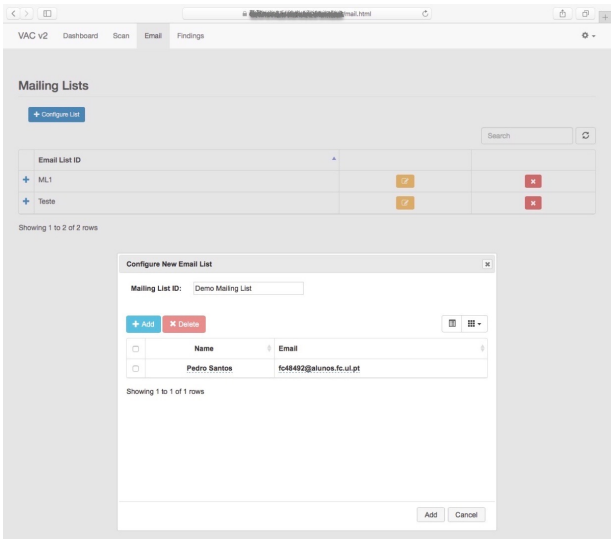


Figure 5.22: Mailing Lists Management

Illustration 5.22 is showing the Mailing List section, and this is one of the most straightforward pages in the app. This image illustrates the creation of a new mailing list, and to achieve that, what needs to be done is to set the Mailing List's Id, and then declare the name and email of the people who are to be notified[9].

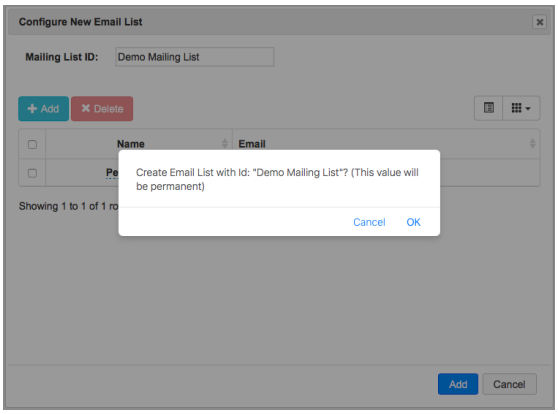


Figure 5.23: Confirmation Popout

Image 5.23 displays a confirmation for the creation of the new Email List. The purpose of this image is to show that, when performing a create, update or delete action, confirmation popouts will be prompt to the user for confirmation of intent.

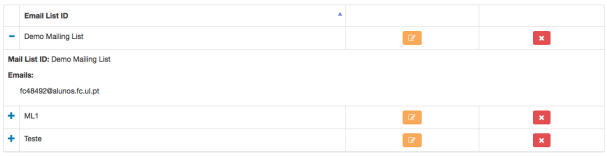


Figure 5.24: Mailing Lists' Table - Detail View

Image 5.24 portrays the Mailing List table and the information that it presents. As the reader might see it is very trivial, showing only the Ids, and when expanded it will present the emails associated with the list.

[9]Although both the Name and Email properties are required, the Name property contains an informational character.

Scan's Page

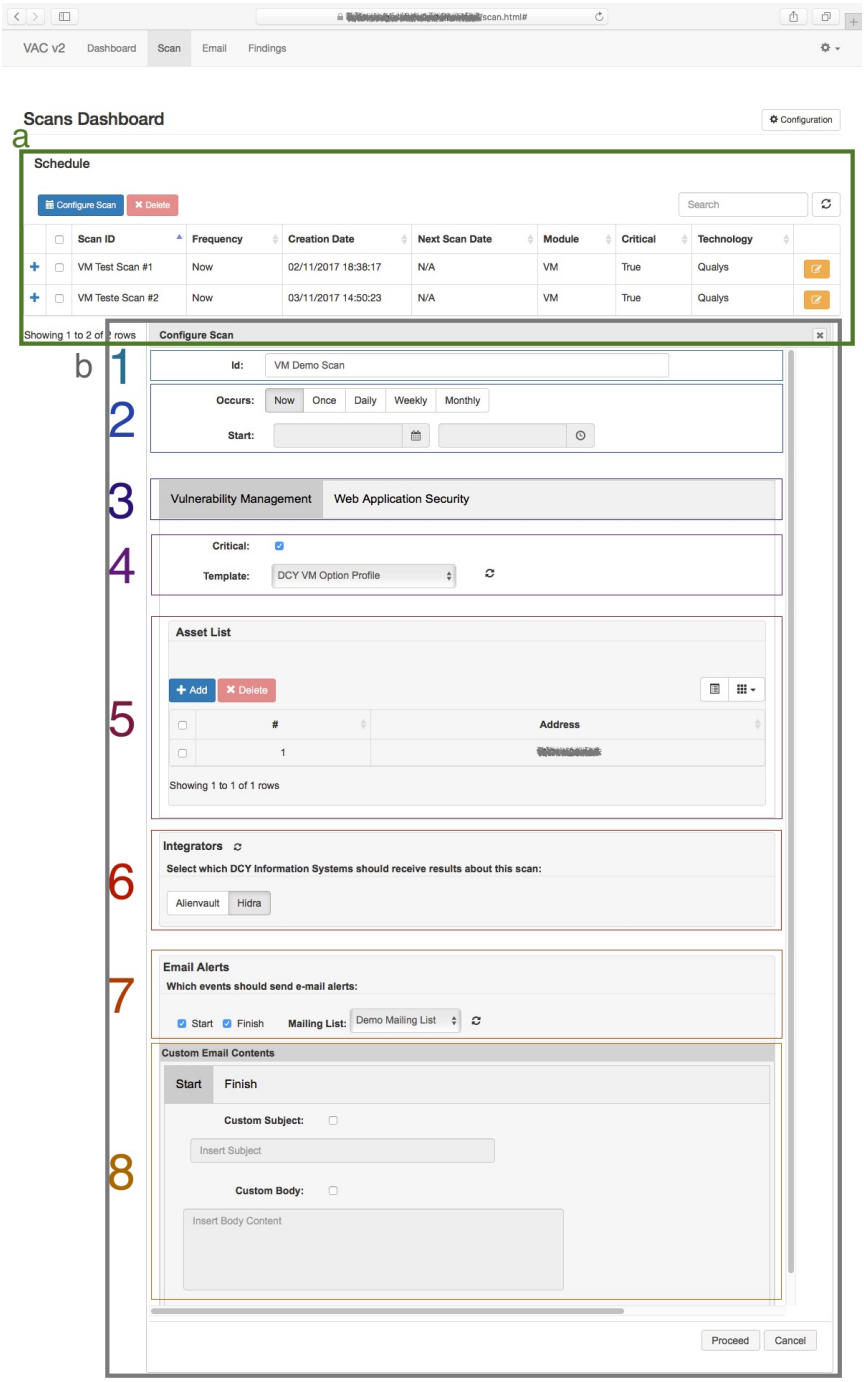


Figure 5.25: Scan's Management & Scan Configuration

Illustration 5.25 depicts the Scan's page. This illustration is showing two screens, represented by the letter "a" is the view we get when accessing this page, and that is composed of a table with the scans configured. The letter "b" represents the view of configuring a

new scan[10], and there are multiple fields to fulfill. A brief description of the fields lies next:

1. Scan Configuration Id - This is the property which will identify the scan, and its occurrences within the platform.
2. Scan recurrence - There are five options, and accordingly to the one that has been chosen it will be shown correspondent fields to fill.
3. Scan type - The tab name represents the scan type which is being configured.
4. Scanning Template - Both the critical and template properties highlighted here, will define the scanning technology that will perform the scan, and with which template.
5. Target's of the scan - This area is where the targets of a VM scan are stated.
6. Integrators - This is related to the third objective of this project, Results treating & Data correlation. For that reason will get back to this later on.
7. Email Alerts - This section will activate notifications for this scan configuration. The operator will be able to choose which states of the scan will trigger a notification. Also, it will have to choose a mailing list to send the notifications.
8. Custom Email - This section provides the operator with the power to customize the title or the body of every notification to be sent.

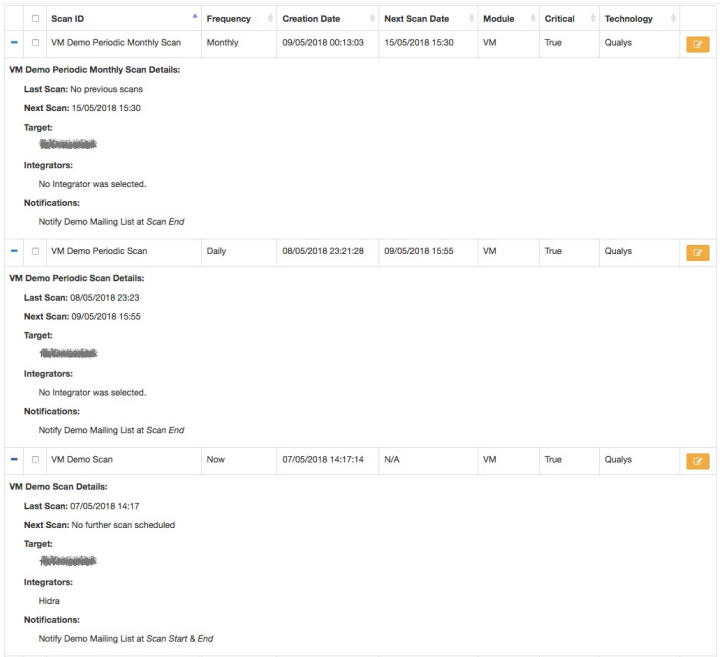


Figure 5.26: Scan Configurations' Table - Detailed View

[10]This is similar to a scan edit screen, the difference is that the scan edit will present the actual values configures in that appliance

Illustration 5.26 depicts the detailed view of the table highlighted by the letter “a” displayed in image 5.25. In this figure, it is presented three different scan configurations.

Let’s analyze the first scan configuration, and see the information that is possible to extract from this table. The first scan is a scan that will occur monthly, and the next scan date will be the May 15th, it is a VM scan, and will be executed by Qualys which is a technology for critical targets. From the additional information we know that this scan is still to occur, the targets of the scan, the integrators were the results are going to be uploaded to, and finally, that this scan will notify the mailing list “Demo Mailing List” at scan start and scan end.

At this point, the reader might have already noticed a yellow button in some of the images presented previously, this is the edit/update button, which will open the corresponding object with the configuration saved in VACv2. This was one of the goals to achieve in this Improvement of VAC objective, and it did make the software jump concerning usability and user-friendliness.

5.2.2 Addition of a new Scanning Technology into VACv2

This section reflects the changes done over VACv2 to achieve the second objective, the addition of a new scanning technology. For VACv2 to interact with Qualys - the new scanning technology - it will be with resort to the Qualys’ API. However, there is no Library provided by Qualys or by Ruby to make use of Qualys’s API.

Qualys scanning technology performs not only Vulnerability Management scan but also Web Application Scanning scans, meaning a new scanning type will be introduced into VACv2.

This section will have a similar structure to the previous one, presenting only the modules affected by this transformation. The difference is that all modules affected are working towards the same goal, instead of attaining smaller goals to achieve the higher objective, as in the previous section.

Scanner Manager Module

Having into consideration the state of this module regarding the first objective, let’s now have a look at the changes made to contemplate the new scanning technology.

From illustration 5.27, the reader can recognize that the only addition to the solution was the Qualys-related classes. Concerning VACv2, the inclusion of the new scanning technology class ran smoothly, Qualys class, this class will handle the proceedings required by the scanning technology to perform the scans, and will also perform the interpretation of the responses provided by Qualys into VACv2.

However, like said before, there was no available resource in Ruby to make use of the Qualys’ API, meaning that a custom library to make requests to Qualys’ platform, and received the responses also had to be developed for VACv2 to become able to communicate

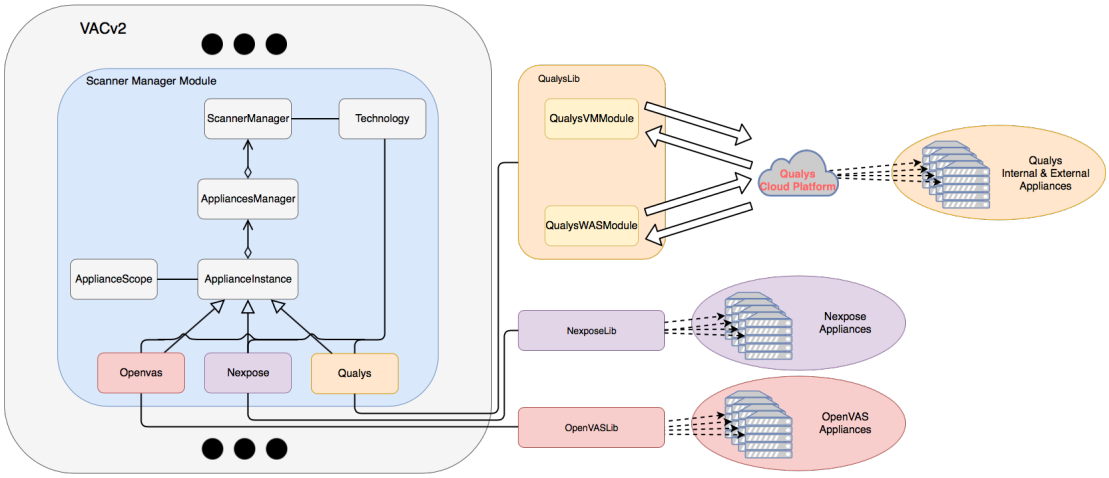


Figure 5.27: VACv2's Scanner Manager module internal structure with Qualys Technology

with the scanning technology. The GET and POST functions made to allow the custom Qualys library to communicate with Qualys' API were built with resort to a Ruby library named "httparty". Regarding authentication, Qualys allows basic authentication, which means that at every request the username and password were sent to Qualys in the request headers.

Let's now have a closer look at the developed classes, and what is pretended by them.

QualysVMModule & QualysWASModule These two classes denote the same objective, a module in Qualys. The first represents the Vulnerability Management module while the second represents the Web Application Scanning module.

Qualys contains several cloud applications, and PT acquired two that are being used in the project: Vulnerability Management (VM) and Web Application Scanning (WAS). The VM is related to infrastructure security by finding assets' vulnerabilities, while WAS is related to web application security, which will explore web applications for vulnerabilities.

QualysVMModule & QualysWASModule might be seen as enabler classes, in other words, these will provide the necessary methods for Qualys class to interact with Qualys platform each in their correspondent module. These are the classes that will allow the Qualys class to perform actions like launching a scan, and so on.

Qualys This class contains the same structure as Nexpose and Openvas, meaning it extends the ApplianceInstance class. Qualys class implements the methods inherited from the ApplianceInstance class, and through these methods, it is able to accomplish the needs of both VACv2 as of Qualys platform. In other words, VACv2 pretends to automate the actual procedures that are done by DCY's personnel, for

this to happen, it will have to handle the technologies by itself, and while doing so it has to accomplish every step required by the scanning technologies, *i.e.*, a scan launch action in VACv2 will coincide with a few steps in the technology. Table 5.1 will illustrate that precisely, what a VACv2 action will correspond in Qualys scanning technology.

VM Scan Type		
VACv2		Qualys Platform
Launch Scan	$\longrightarrow$	Add Host Addresses
	$\longrightarrow$	Create Asset Groups
	$\longrightarrow$	Launch Scan
Scan State	$\longrightarrow$	Get Scan Information
Launch Report	$\longrightarrow$	Launch Scan Report
Report State	$\longrightarrow$	Get Report Information
Download Report	$\longrightarrow$	Download Report

Table 5.1: Match of VACv2's actions in Scanner Manager module internal structure with Qualys Technology

Let's now dissect two of VACv2's actions stated in the previous table: the Launch Scan and the Download Report.

- The launch scan action in VACv2 corresponds to three actions in Qualys, and this is because Qualys requires three steps before launching a scan.
  1. The targets of the scan have to be stated in the platform. Otherwise, for security reasons, Qualys will block any scans to IP addresses not stated to prevent accidental scans.
  2. This step is the aggregation of IP addresses with the same characteristics. As previously said, PT has its internal network divided into multiple subnetworks, and a service typically contains IP addresses scattered over multiple internal networks. Qualys contains an object named Asset Group, which aggregates IP addresses, and allows the association of a scanning appliance to that Asset Group, ensuring that the assets contained in the Asset Group will all be scanned by that appliance. In other words, this object will associate the IP addresses contained in a given internal network with an appliance acting in that same network.
  3. Finally, the last action is precisely the launch of the scan, in this step, all the Asset Groups must be referenced, which means that only a scan is needed to be created instead of performing a scan to every Asset group. Also, this is where the scan template will be defined.

- Let's now analyze the Download Report action, this action from the Qualys platform has nothing else to add. However, that is not the case with VACv2. In more detail, Qualys class is the only one with the knowledge of the report structure sent by Qualys. Qualys class is responsible for mapping the results into an internal object - ReportInstance. The ReportInstance object was made to personify any scanning technology report in VACv2. However, as it makes part of the third objective of this project, we will get into more details ahead.

Scan Manager Module

Having into consideration illustration 5.9, which introduces the changes done over this module after the development made to achieve the first objective, let's now have a look at image 5.28 that presents the changes made to achieve the second objective of this project. The addition of the Qualys scanning technology made changes to this module due to Qualys containing one other scanning module that VACv2 was not yet ready to handle.

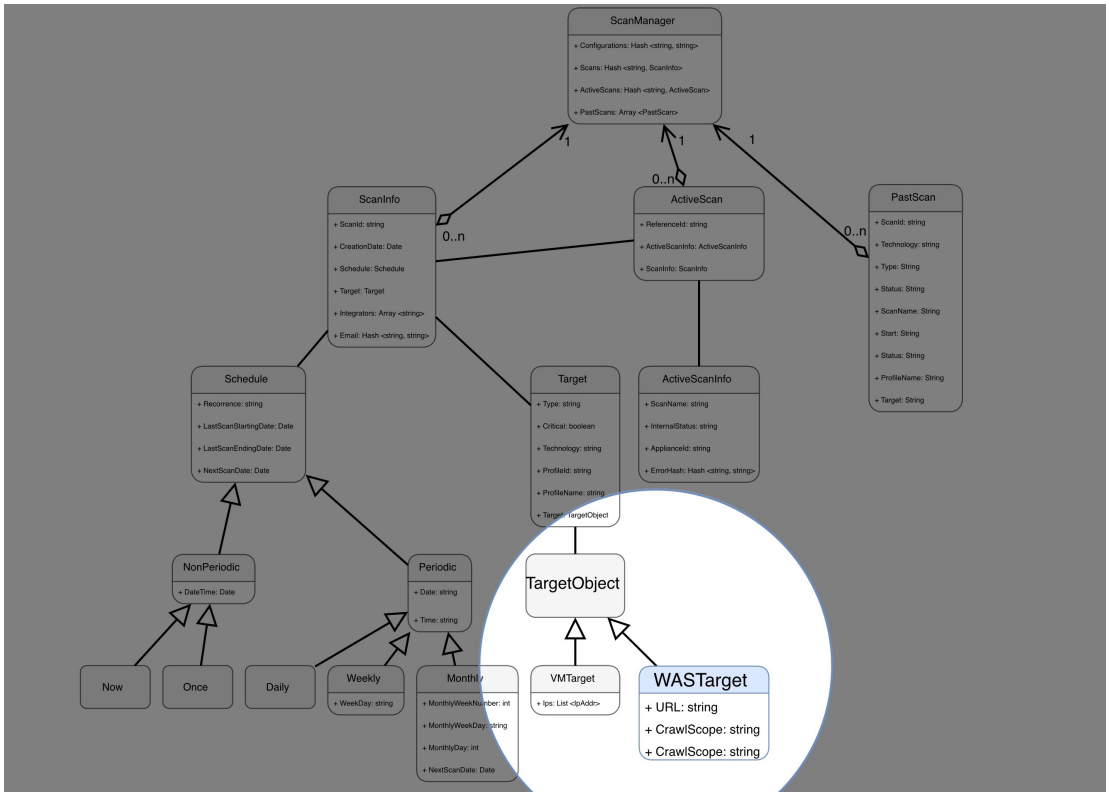


Figure 5.28: VACv2's Scan Manager module internal structure with WAS Target class

From the highlight present in the image, it is possible to observe that the only change made to this module was the addition of a new class named WASTarget that will extend the TargetObject class. The effort in adding this new scan type made possible to acknowledge that at this point VACv2 is quite easier to add new scan types. Let's now get into the detail of this new WASTarget class.

WASTarget This class is equivalent to the VMTarget class, meaning it will extend the TargetObject class.[11] Otherwise, the Target class would have to contemplate every property associated with every single one of the scan types. In other words, that class would have to suffer a change every time a new scan type was to be added, something that did not occur in this case.

The purpose of this class is to hold to every property that is required to perform a WAS scan. This class is composed of three properties, which are the following:

URL Target of the scan, equivalent to the IP addresses in the VMTarget.

CrawlScope This property is intrinsic to WAS scans. It defines the scope of the scan, *e.g.*, should the scan be executed over the domain, should it only be executed over specific sub-domains, *et cetera*.

ApplianceId This property reflects the appliance designated to perform the scan. Although the association is done in runtime[12], in opposition to what happens with a VM scan, the operator has the final word about which appliance will perform the scan. The reason why this happens is that some websites may resolve both to an internal address of PT, as it may resolve externally, so this provides the power for the user to choose the network appliance where the website is located case the original suggestion is not the pretended one.

Auxiliary Module - Network Manager

For this objective, and due to the Qualys scanning technology providing WAS scans, it was needed for this module to resolve an URL to its IP address, and then to associate it with a network. Image 5.29 presents the structure of the file, which is loaded into memory by this module. This module achieves this by associating DNS servers with a custom network. The way it works is by trying to resolve the URL into the IP, if not successful it will move on until one is successful, if none of the DNS servers determines the IP address, then it is assumed that the URL provided is on the internet, being external to PT.

[11] Quick reminder, the TargetObject class exists with the purpose of detaching VACv2 from being attached to a scan type.

[12] By runtime it is meant while the user is still configuring the object of the scan.

Figure 5.29: VACv2's Network Manager Boot File

View Manager Module & ClientSide

Every view presented in the previous section showed VACv2 in its final state, in this section let's analyze the properties involved with a WAS scan while configuring a scan. The following illustrations will exemplify the usage when configuring a WAS scan, and these images will only illustrate the specific WAS properties given that every other option remains the same.

Before starting, if the reader might recollect figure 5.27, which illustrated the configuration of a VM scan in VACv2, what defined that the configuration being done was a VM scan instead of a WAS scan, was the tab chosen. Let's now take a look at the Web Application Security tab.

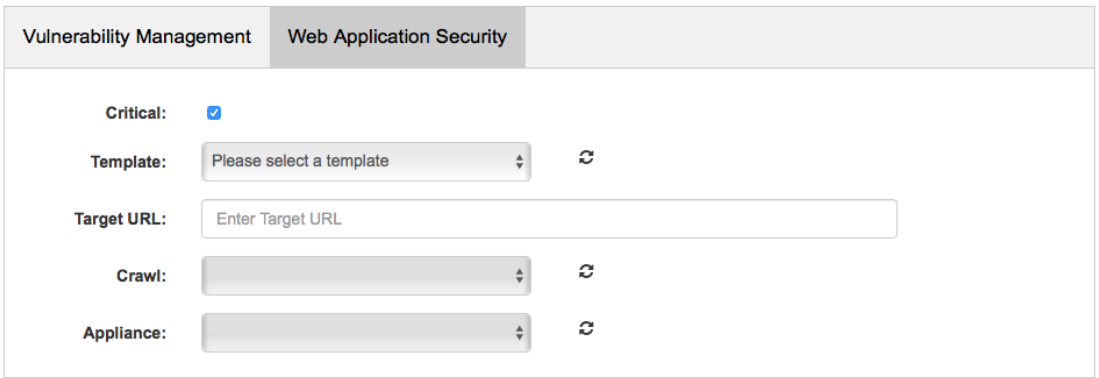


Figure 5.30: WAS Scan Configuration Tab illustration

Image 5.30 illustrates the tab related to the WAS configuration. Please note that the Crawl and Appliance options are blocked.

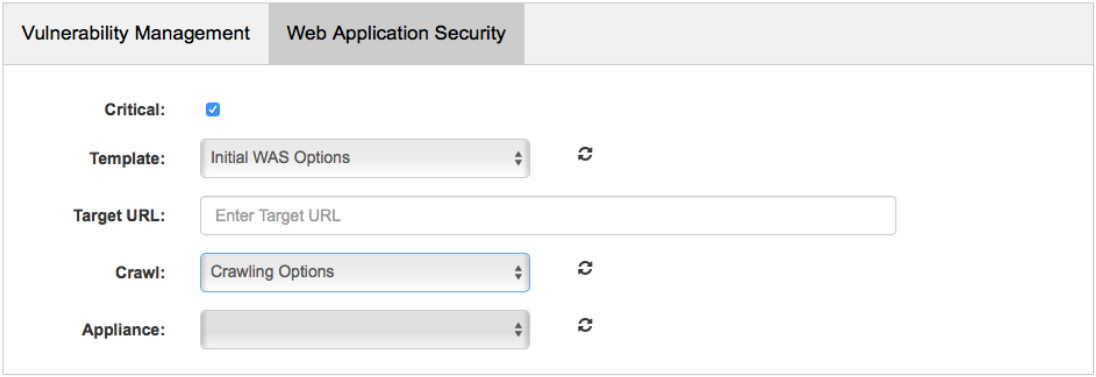


Figure 5.31: WAS Scan Configuration Tab illustration - First Step

Image 5.31 is now illustrating that after choosing the template, the crawl option gets unlocked. Every scanning technology may provide its crawling options, that is why the option is blocked until then because the template is an indirect pointer to the scanning technology.

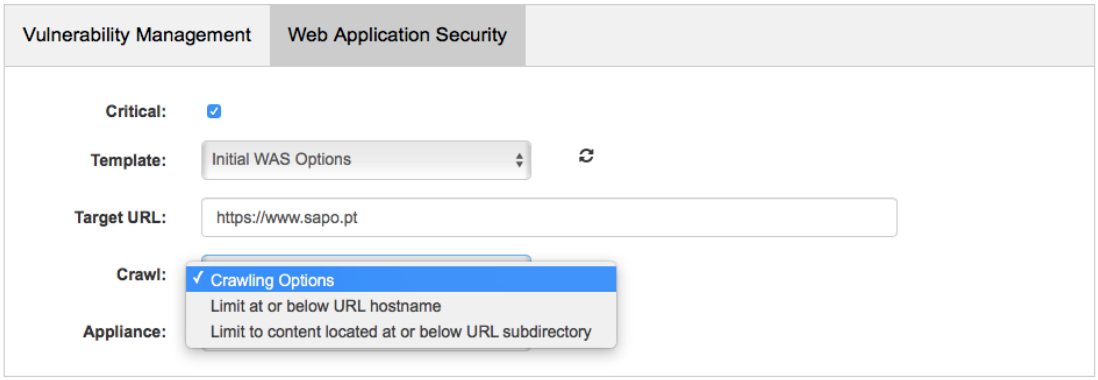


Figure 5.32: WAS Scan Configuration Tab illustration - Second Step

Image 5.32 illustrates the available crawling options for the Qualys scanning technology. It is possible to infer that is the Qualys technology because at this time only Qualys has the possibility to perform WAS scans. Otherwise, the user can be aware of the actual scanning technology when taking a look at the template property.

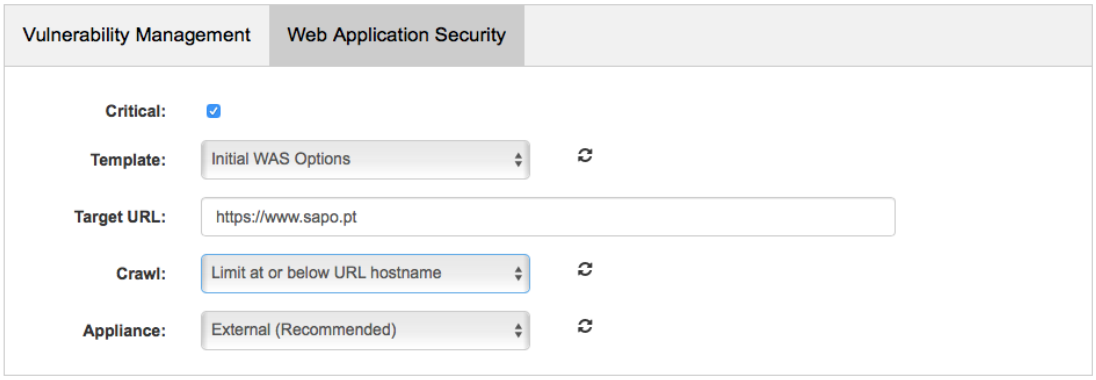


Figure 5.33: WAS Scan Configuration Tab illustration - Third Step

Image 5.33 is illustrating the automatical assign of an appliance to perform the scan. The Appliance field only gets unlocked after choosing the Template option and inserting the Target URL. It works this way because, we need to know which scanning technology is going to perform the scan so we can get the networks currently monitored by its appliances, and then the Target URL is needed for the DNS servers to try to resolve it, and this way associate the URL with an appliance.[13]

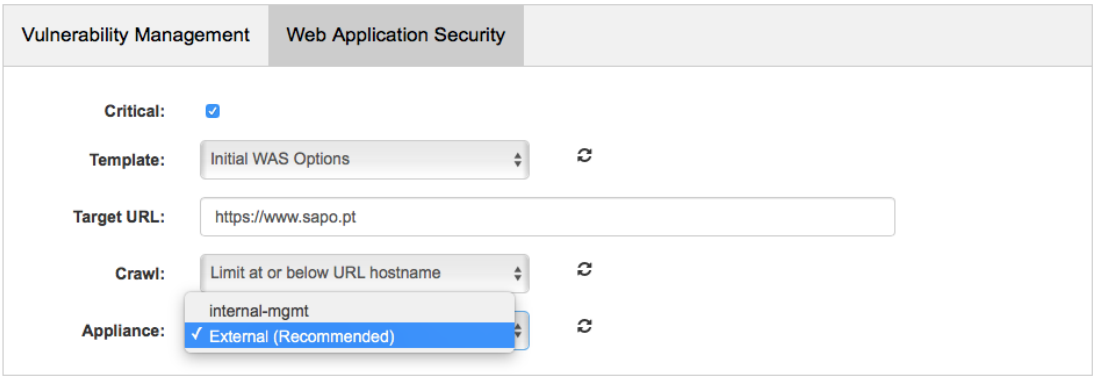


Figure 5.34: WAS Scan Configuration Tab illustration - Fourth Step

Image 5.34 illustrates the available networks where Qualys can perform scans, and which of the appliances was the suggested for performing the scan.

5.2.3 Results Treating & Data Correlation

This section will describe the third and final objective of this report. This objective could be split into two phases. The first phase consists of the redesign of VACv2's Results Manager module, while the second, in a very high-level detail, would consist on Maltego getting the scan results which were placed in Hidra.

[13]This is the illustration of the feature mentioned in the section 5.2.2 Scan Manager Module while describing the WASTarget class' ApplianceId property.

The changes made over the Results Manager module had the intention of providing the operator control of VACv2' results. The second part takes place in Maltego - a specialized software for data correlation -, it will consist in a development over the software, for Maltego to be able to fetch the results of VACv2 from Hidra. The aim for this "second phase" is to allow an easier correlation of data, which will from now on, include vulnerability scan results. The operator of Maltego is not the same as the operator of VACv2, Maltego is to be managed by a SOC operator.

Following, this section will have a similar structure to the previous ones, presenting only the affected modules by the change.

Results Manager Module

This module was not mentioned previously due to having suffered a complete redesign face what was previously done in VAC, and the reason why is that VAC was not able to handle any interaction or customization over this module. In fact, the classes that were used to connect with the DCY's information systems - designated by integrators -, contained all the sensitive information belonging to the repositories tangled up with the coding itself, information like host address, the username or the password.

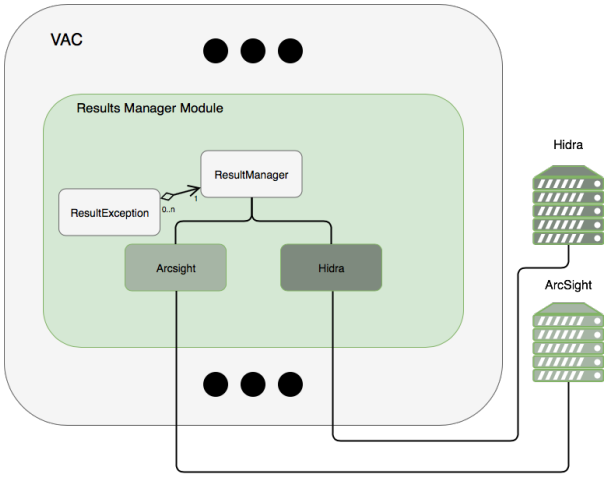


Figure 5.35: VAC's Results Manager module internal structure

Figure 5.35 illustrates the architecture of this module over VAC, it is possible to witness that the ResultManager class had an association with the integrators, restraining to a one-to-one relationship. It is also possible to recognize that the system was not ready for handling technologies besides the two that already existed, at least without having a development. The ResultManager class also stored the events signaled to except in future scans, this information was kept in an array of ResultException classes. Typically, every ResultException class represented a false-positive event identified in the scannings.

In VAC, this module used to be triggered through the creation of files that transmitted

that a scan had just finished, the scan results would be the content of such file. This module would open that files, load the content into memory and sent the information directly to the integrators' classes. The ArcSight and Hidra classes would parse the results for excepted events, and then upload the results into the information repository. At every scan, ArcSight and Hidra would perform their routines, meaning the results of every scan would be contained in these two systems, something not desired.

Let's now analyze the changes the Results Manager module suffered from the implementation of the third objective of this project.

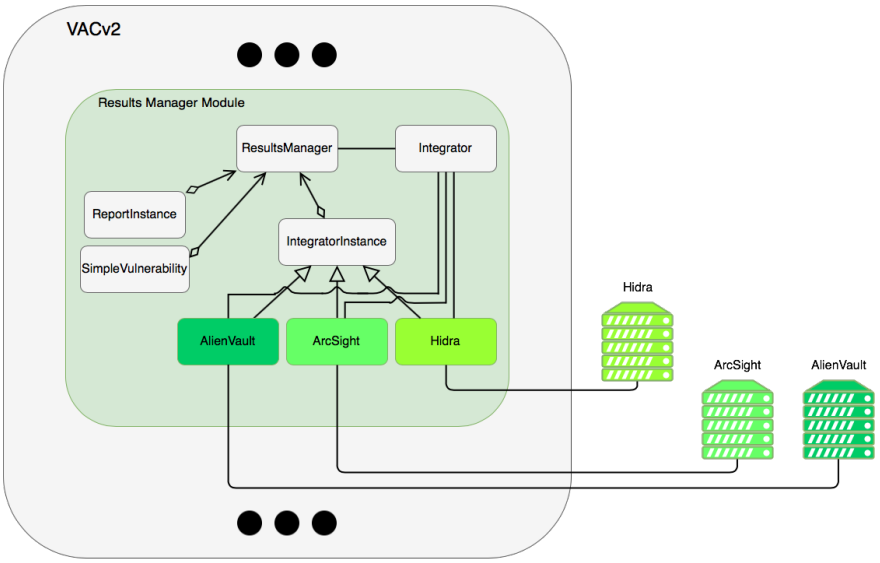


Figure 5.36: VACv2's Results Manager module internal structure

As it is possible to understand just by comparing illustration 5.36 with illustration 5.35, this module has raised quite a bit. The structure that this module now has was based on the Scanner Manager module, hence the resemblance, *e.g.*, the Scanner Manager module's Technology class *vs.* the Integrator class, or the ScannerManager module's ApplianceInstance class *vs.* the IntegratorInstance class. Let's analyze the classes illustrated in image 5.36 in more detail.

Integrator Although with a different context, this class aims at the same purpose as the Technology class in the Scanner Manager module. Its purpose is to recognize new technologies for the Results Manager module, store them, and to load them as well as their properties. However, in this case, the technologies recognized will be related to DCY's Information Systems. The properties are being mentioned because they are dynamic, and they will appear in the client-side according to what is stated in the configuration files.

IntegratorInstance The same happens with this class, with a different context, but this class aims at the same purpose as the ApplianceInstance class in the Scanner Man-

ager module. This class's purpose is to be a single point of unity for the VACv2 software and the classes impersonating the DCY's Information Systems. This class will provide a base structure for the classes impersonating the Information Repositories, ensuring an abstract layer between the ResultManager class and the classes extending this one, this way strengthening VACv2's unattachment to any technologies. This class contains a few properties, but only one is required to be explained, it is a hash named configurations. What will be stored in that hash are the properties for the technology that it is impersonating. Ahead, will be an illustration to show the importance of the properties in the client-side, to make more accessible for the reader to understand.

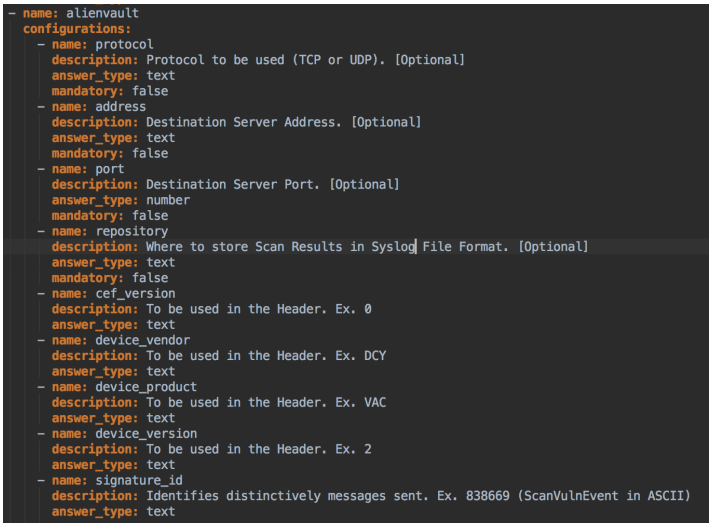


Figure 5.37: VACv2's AlienVault integrator properties

Illustrated in image 5.37 is the configuration belonging to the AlienVault integrator in VACv2. From it is possible to observe multiple properties and that all follow a four attributes structure, these attributes have the purpose of instructing VACv2 how to handle the properties. These properties presented in the figure are the ones that will be stored over the configurations' hash mentioned previously, along with its value.

AlienVault, ArcSight & Hidra These classes are the ones that impersonate a DCY's Information System. They will be extending the IntegratorInstance class and implementing its base structure. Also, they are responsible for using the configurations' hash which contains the properties each technology requires to achieve a successful connection to its information repository. Each one of these classes will upload data according to the format the technology is expecting. From the three classes, AlienVault is the only brand new technology to integrate, but Hidra as suffered changes,

so we will get into more detail over the way these two upload its values into its Information System.

AlienVault It was defined that the integrator for AlienVault would upload its results through Syslog [34], which is a standard for message logging, and would use the CEF format[47]. CEF stands for Common Event Format, which defines that the syslog message should contain the following structure:

CEF: Version|Device Vendor|Device Product|Device Version|
Signature ID|Name|Severity|Extension

Ahead, in the Results chapter, there will be an opportunity to view a real event sent by VACv2 to AlienVault. However, based on the CEF structure, it is possible to say that the "Extension" field contains all the information related to the event, all the other fields are related to the software, in this case, it will contain information about VACv2.

Also, in this integrator, only the Vulnerabilities will be sent to the technology and one by one, this has much to do with the fact that AlienVault is a SIEM. In other words, it means that the called "Metadata" stored in the ReportInstance's summary property will not be sent to AlienVault.

Hidra As said before, Hidra is a custom solution, and it contains an in-house Ruby library for uploading values into the system. Hidra is composed of multiple software, but it is centralized in the ElasticSearch Engine. ElasticSearch uses indexes to store data, the definition of an index by Elastic is as follows:
"An index is like a table in a relational database. It has a mapping which contains a type, which contains the fields in the index..." [27]
With this in mind, VACv2 uploads the values of a scan into two indexes, they are named "scan_event" and "vuln_event", both with well-defined purposes.

The "scan_event" will contain the called Metadata of the scan, *i.e.*, will contain information like the number of vulnerabilities found, the average response by the target, or the total hosts number. Image 5.38 illustrates a table with the fields this ElasticSearch index will be composed, let's now see what each column of the table will provide:

- First column refers to the name of the field in ElasticSearch;
- Second column states with which scan type the property is related;
- Third column details which field of the ReportInstance is responsible for filling this property. In some cases, the ElasticSearch property can have more than one field according to the scan type;

- Fourth column is a brief description of the ReportInstance field.

ElasticSearch Index: scan_event

ElasticSearch	Scan Type	VACv2's ReportInstance Field	Description
action_result	VM & WAS	Summary.ScanStatus	Scan indicator of problems
action_result_attachment	VM & WAS	PdfReport	PDF Scan Report
object	WAS	Summary.Target.Target	Target URL
object_ip	VM & WAS	[VM] Summary.Target.Target [WAS] Summary.Target.TargetIP	[VM] Targets of the scan [WAS] IP Address of target URL
object_os	VM & WAS	Summary.Target.TargetOS	Operating Systems of the Targets
scan_assess_duration	WAS	Summary.Target.AssessDuration	How much time the appliance took to find all the links
scan_avg_response	WAS	Summary.Target.AvgResponse	Average response of the targe
scan_crawl_duration	WAS	Summary.Target.CrawlDuration	How much time the scanning appliance take to crawl all the links
scan_crawled_links	WAS	Summary.Target.CrawledLinks	Links crawled by the scanner appliance
scan_hosts_dead	VM	Summary.Target.TotalHosts - Summary.Target.HostsAlive	Host not Alive
scan_hosts_total	VM	Summary.Target.TotalHosts	Total hosts
scan_name	VM & WAS	Summary.ScanName	Scan Name
scan_total_duration	WAS	Summary.ScanDuration	Duration of the Scan
scan_total_links	WAS	Summary.Target.TotalLinks	Total links discovered in target
scan_total_requests	WAS	Summary.Target.TotalRequests	Total requests made to the Target
source	VM & WAS	Summary.ScanType	Scan type
technology	VM & WAS	Summary.ScanTechnology	Scanning Technology
ts	VM & WAS	Summary.ScanStartingDate	Scan starting date
vuln_found	VM & WAS	Summary.VulnerabilitiesFound	Number of Vulnerabilities Found

Figure 5.38: ElasticSearch Index "scan_event" *vs.* VACv2's ReportInstance Structure

Let's now analyze the "scan_vuln" index, which purpose is to contain every event reported by the scanning technologies. Like the "scan_event", image 5.39 is illustrating a table, which as the same structure as the previous one. However, this table does not contain the description field, given that the fields present in this table refer to common fields related to vulnerabilities.

ElasticSearch Index: scan_vuln

ElasticSearch	Scan Type	VACv2's ReportInstance Field
action_details	WAS	Vulnerabilities.Vulnerability.TargetVuln.Auth || Vulnerabilities.Vulnerability.TargetVuln.Ajax
external_id	VM & WAS	Vulnerabilities.Vulnerability.Id
object	WAS	[VM] Vulnerabilities.Vulnerability.TargetVuln.OSCPE --- [WAS] Vulnerabilities.Vulnerability.TargetVuln.Target
object_details	VM & WAS	[VM] Vulnerabilities.Vulnerability.TargetVuln.Protocol --- [WAS] Vulnerabilities.Vulnerability.TargetVuln.TargetURL
object_ip	VM	Vulnerabilities.Vulnerability.TargetVuln.Target
object_port	VM	Vulnerabilities.Vulnerability.TargetVuln.Port
object_service	VM	Vulnerabilities.Vulnerability.TargetVuln.Service
scan_name	VM & WAS	Summary.ScanName
source	VM & WAS	Summary.ScanType
technology	VM & WAS	Summary.ScanTechnology
ts	VM & WAS	Summary.ScanStartingDate
vuln_class	VM & WAS	Vulnerabilities.Vulnerability.Category
vuln_code	VM & WAS	[VM] Vulnerabilities.Vulnerability.TargetVuln.CVE || Vulnerabilities.Vulnerability.TargetVuln.PCI --- [WAS] Vulnerabilities.Vulnerability.TargetVuln.OWASP || Vulnerabilities.Vulnerability.TargetVuln.WASC || Vulnerabilities.Vulnerability.TargetVuln.CWE
vuln_description	WAS	Vulnerabilities.Vulnerability.TargetVuln.Description
vuln_exploit	VM	Vulnerabilities.Vulnerability.TargetVuln.Exploit
vuln_impact	VM & WAS	Vulnerabilities.Vulnerability.Impact
vuln_last_detection	VM & WAS	Vulnerabilities.Vulnerability.LastDetection
vuln_malware	VM	Vulnerabilities.Vulnerability.TargetVuln.Malware
vuln_payload	WAS	Vulnerabilities.Vulnerability.TargetVuln.Payload
vuln_policy	VM & WAS	Vulnerabilities.Vulnerability.Policy
vuln_policy_description	VM & WAS	Vulnerabilities.Vulnerability.PolicyDescription
vuln_ref	VM	Vulnerabilities.Vulnerability.TargetVuln.Refs
vuln_severity	VM & WAS	Vulnerabilities.Vulnerability.Severity
vuln_solution	VM & WAS	Vulnerabilities.Vulnerability.Solution
vuln_status	VM	Vulnerabilities.Vulnerability.TargetVuln.Status
vuln_test_result	VM	Vulnerabilities.Vulnerability.TargetVuln.TestResult
vuln_threat	VM	Vulnerabilities.Vulnerability.TargetVuln.Threat
vuln_title	VM & WAS	Vulnerabilities.Vulnerability.Title
vuln_type	VM & WAS	Vulnerabilities.Vulnerability.Type

Figure 5.39: ElasticSearch Index "scan_vuln" *vs.* VACv2's ReportInstance Structure

Both indexes contain the structure that made the most sense to DCY and the student.

ReportInstance The ReportInstance class is the object that will hold all the information related to the results of a scan. As there are multiple scanning technologies, and at least two different scan types, this class must be able to support all the different reports produced by the technologies. Image 5.40 is illustrating the composition of the Report Instance object.

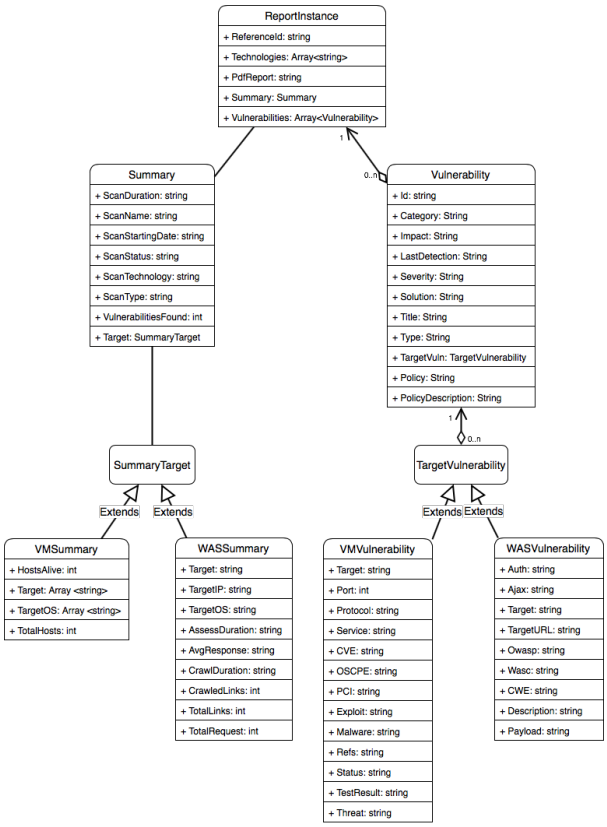


Figure 5.40: ReportInstance Structure

All the fields present in the structure were chosen as being the most "regular" between technologies. What this intents to achieve, is a way to get VACv2 detached from any technology, being able to parse any report from any scanning technology. The creation of this object is done in the classes that extend the ApplianceInstance class, due to these classes being the ones who will contain the knowledge of how the results should be mapped into the ReportInstance object.

The ReportInstance is composed of two children objects, let's have a high-level analysis of what is pretended from these objects:

Summary This object's purpose is to contain the information that might be seen as "Metadata" of the scan, meaning it will contain the name of the scan, how much time did the scanning take, the targets of the scan, the number of hosts alive *vs.* the number of total hosts, the number of vulnerabilities found, and so on.

Vulnerability This object contains the purpose of carrying the information about the event and the target where it was found.

There are only two fields that belong to VACv2 and not to any of the scanning technologies, which are: "Policy" and "PolicyDescription". Both situated in

the Vulnerability class, these fields will store the value if the vulnerability has been excepted or excluded in the VACv2 system, with a brief description from the user to explain the reason why, respectively.

SimpleVulnerability This class's purpose is to impersonate a vulnerability of a given technology. It will store the values that uniquely identify any vulnerability in any scanning technology, properties like the scanning technology, the identifier of the vulnerability in the scanning technology, the title of the vulnerability, and so one.

ResultsManager This class is the head of the Results Manager module. Its purpose is to, in a first instance, manage all the instanciated IntegratorInstance classes, secondly to parse every scan result, while storing every unique vulnerability from each technology[14], and finally sending the parsed ReportInstance into the correspondent integrators for uploading the results into the platforms.

This class is composed of the following properties:

- IntegratorsInstance's hash
- ReportInstance's Hash - These objects are stored by state, which represent the phase were the ReportInstance is. *E.g.*, Requesting Report, or Downloading Report, or Parsing Results, or Uploading Results.
- SimpleVulnerability's Hash - This property contains every event ever reported by any scanning technology.
- ExceptedSimpleVulnerability's Hash - This property will contain every event which was excepted. This property will point to the SimpleVulnerability object which represents the vulnerability and will associate an IP address and a motive.
- ExcludedSimpleVulnerability's Hash - This property will contain every event which was excluded. This property will point to the SimpleVulnerability object which represents the vulnerability and will associate a motive.

The ExceptedSimpleVulnerability and ExcludedSimpleVulnerability properties refer to two different actions that might be carried by the operator.

- An excepted event is so when a given event is associated with a target, meaning that it is to be considered as a false-positive. Let's imagine the following example: A given target contains the port associated with the FTP protocol open. However, it is not the FTP service that is active and is one other protocol that is secure. The result would be this event being excepted for this target, for this vulnerability, and in this service/port.

[14]The vulnerabilities are saved in the SimpleVulnerability object.

- An excluded event is so when a given event is to be excluded from any scan configuration, meaning this event does not need to be associated with a target. The purpose of this action is to if such event is ever found to be ignored despite the target. Let's imagine the following example: One scan reported an event in which the subject common name of the certificate did not match the server FQDN. If VACv2 were to be only used for scanning internal IP Addresses, and given that external certificates cannot contain internal addresses, that would result in the exclusion of this vulnerability from the system.

This module's normal operation is as follows: In reverse to what used to happen in VAC, VACv2's Results Manager module will be triggered by the Scan Manager module. When a scan reaches its ending, the Scan Manager module will verify if in the scan configuration it was configured at least one IntegratorInstance identifiers. If so, would be stored in the integrators property, mentioned in section 5.2.1. For the scans whose integrators property is not empty, the Scan Manager module will request this module to take action.

On the Results Manager Module side, when receiving the request by the Scan Manager module, this module will request the Scanner Manager module to generate a scan report for that scan. The Results Manager module will track the report generation through inquiries to the Scanner Manager module, and when it reaches the end, the Results Manager module will request the Scanner Manager module to download the report, which will then interpret the results, map them into a ReportInstance object, and return it to the Results Manager Module.

When this module gets the ReportInstance object, it will iterate all the results for new events, and to signal events that might be excepted or excluded. The way it is done is by setting the "policy" property with the values "excepted" or "excluded" and filling the "policydescription" value with the information contained in the ExceptedSimpleVulnerability or ExcludedSimpleVulnerability hash. After the results being checked, the ResultsManager class will request the upload of the ReportInstance to the Integrators which IDs were referenced in the ReportInstance's Technologies[15] array.

Scan Manager Module

At this point, this module is here to explain a single feature that was already shown but not explained at the time. If the reader might recall figure 5.25, which consisted of the illustration of a scan configuration, available in section 5.2.1, while analyzing the differences made in the client-side over the first objective of this project - Improvement of VAC. The integrators property of the ScanInfo object carries the purpose of containing the chosen integrators, which point to the DCY's Information Repositories where the scan results are

[15]The technologies in this context are a reference to the integrators.

to be upload. The values contained in this array will be the ResultsManager's integrators Ids.

View Manager Module & ClientSide

This section will reveal the modifications done over the client-side while in the scope of the third objective of this project. For the data related to the Results Manager module to get to the client-side, an association between this module and the View Manager had to be made, similarly to the other modules already illustrated in figure 5.10.

Image 5.41 will illustrate the page related to the Results Manager module view in the client-side of VACv2, and there we can see the available integrators, which are loaded once the Results Manager module recognizes them.

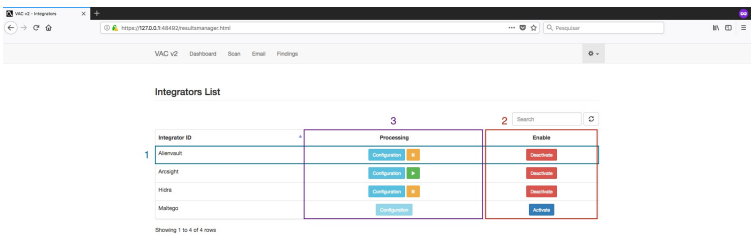


Figure 5.41: Integrators Management

Let's analyze the three highlights in the image:

One Exemplifies the integrator for the AlienVault technology. From this row, it is possible to view that the technology is "enabled" and "processing".

Two Pretends to explain the technology state for the different states of the button available:

- Button "Activate" - This means that the technology is currently disabled and not configured. In this state, the user is not allowed to change any configuration to this technology. In fact, the Configuration button is locked.
- Button "Deactivate" - This means that the technology is currently enabled, which will allow the configuration of the technology. When the operator presses this button, this will trigger not only the deactivation of the technology but will also erase all configurations associated with the technology.

Three Pretends to demonstrate the different "processing" states associated with a technology:

- Configuration button locked - The technology is deactivated, and has to be activated before being possible to associate it with a configuration[16].
- Configuration button with Resume button - VACv2 will only allow the operator to click the resume button when the configuration is set.

  The resume button is what will manifest that the technology is now ready to use. In more detail, only when this button is clicked is that this integrator will start appearing in the scan configuration page.
- Configuration button with Pause button - This means that the technology is active, configured and currently running. Only the technologies with this state will be shown in the scan configuration page.

Illustration 5.42 is a clip of illustration 5.25 - previously shown -, it is illustrating the active, configured and currently running integrators. In this image, the user must select or deselect according to where the scan results are to be uploaded. The illustration only presents two integrators, which if the reader might recall were the same as the ones running, as shown in illustration 5.41.

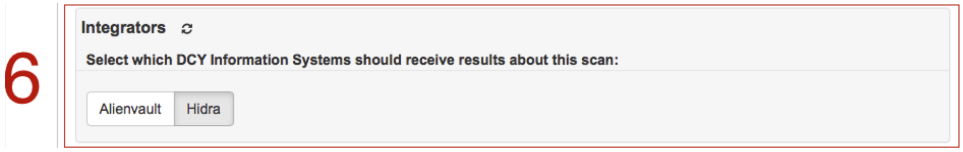


Figure 5.42: Scan Configuration Integrator's property

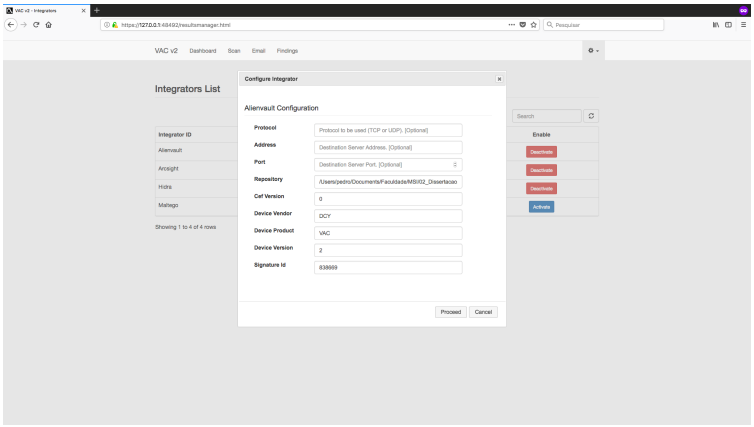


Figure 5.43: AlienVault Integrator properties

Figure 5.43 is illustrating the configuration of the AlienVault technology in the client-side. This configuration was shown with the purpose of enlightening the reader on how

[16]Illustration 5.41 contains the Maltego technology for demonstration purposes only, and the row is currently in this state.

VACv2 will handle the integrator's properties, this illustration is the client-side associated with the figure 5.37 previously shown. It is possible to observe that the configuration file, which defines the properties, is also defining how the properties will be presented in the client-side, from the property type to the tooltip, or even if they are mandatory or not.

Maltego

As already mentioned, the original intent for Maltego proved to be unfeasible due to software compatibility problems. The solution to the problem was to make VACv2 upload the results to Hidra, and then make Maltego fetch the results from this information repository.

For this solution to work, Maltego has to fetch the pretended scan results, from ElasticSearch with resort to its API. As at the time, Maltego did not offer any support for the Ruby programming language, Ruby had to be set aside. The chosen language was one of the suggested by Paterva - creators of Maltego -, Python[31]. Python also had a framework, named Canari[23], developed explicitly for creating Maltego transforms[17].

So it was decided that this "second phase" of the third objective was going to be developed in Python programming language, version 2.7. To retrieve any information from ElasticSearch, it was required the implementation of a custom library for handling the connection and the searches over this platform. It was achieved through the use of ElasticSearch's API, and a custom library developed for the purpose was named ElasticConnector. Image 5.44 illustrates the association of the ElasticConnector with the remaining classes.

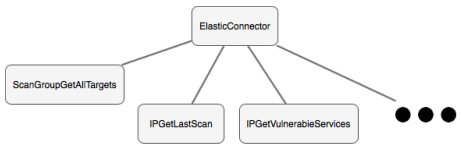


Figure 5.44: Maltego Transforms structure

Illustration 5.44 is presenting the ElasticConnector library, associated with some other entities, which represent Maltego transforms. As there are about thirty-eight transforms these were omitted from the image.

When starting the development of the Maltego's transforms, it was noticed that the default entities that already existed in the software were not enough to represent the scans' data. The alternative was to created custom entities. Image 5.45 presents all the entities created for this purpose.

[17] A transform can be seen as a procedure/ method in any other language.

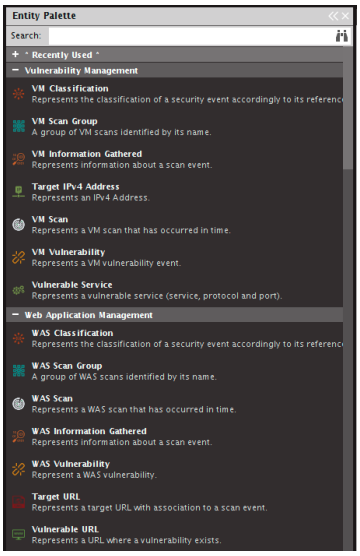


Figure 5.45: Maltego custom entities for handling VACv2 data.

Besides the entities presented in the previous illustration, there is one more belonging to Maltego which was used, named IPv4 Address. Next, it will be described the purpose of each of these entities.

IPv4 Address - Represents a simple IPv4Address with no attachments.

VM/WAS Scan Group - It represents a scan configuration on VACv2.

VM/WAS Scan - Represents an instance of a scan configuration.

Target IPv4 Address - This entity represents an IPv4 address that is associated with a VM Scan entity.

Vulnerable Service - This entity represents a service which was reported in the scanning technology and is associated with an IPv4 Address or a Target IPv4 Address.

VM Vulnerability - This entity pretends to represent a potential vulnerability or a real vulnerability associated with a Vulnerable Service.

VM Information Gathered - Because Information Gathered events usually are not that relevant, they are in a separate entity to allow an easier data manipulation. They are associated with a Vulnerable Service entity, typically is the one containing the value “N/A”.

VM Classification - This entity will represent a CVE associated with a VM Vulnerability.

Target URL - This entity is similar to the Target IPv4 Address in the VM context. However, as in the WAS scans we do not scan an IP address but URLs, this will be the targeted URL.

Vulnerable URL - This entity is similar to the Vulnerable Service in the VM context. However, when performing a WAS scan, the URL contained by this entity will be the vulnerable one.

WAS Vulnerability - Equal to the VM Vulnerability with the difference that it applies to a Vulnerable URL.

WAS Information Gathered - Equal to the VM Information Gathered with the difference that it applies to a Vulnerable URL.

WAS Classification - This entity will represent a CWE associated with a WAS Vulnerability.

After analyzing the entities and their purpose let's move onto the developed transforms. There will be a brief description of every one of the transforms contained in illustration 5.46. A transform contains a signature like methods in programming languages, and will only show when at the correct input entity.

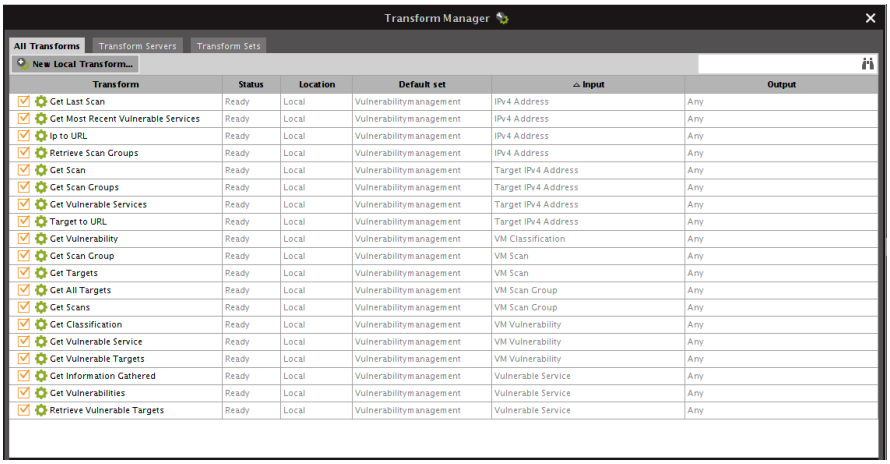


Figure 5.46: Maltego transforms concerning VM Scans.

Get Last Scan - The input for this transform is the IPv4 Address entity. This transform will search for the most recent VM Scan where this IP address has been one of the targets.

Get Most Recent Vulnerable Services - The input for this transform is the IPv4 Address. This transform will search for the most recent Vulnerable Services associated with the IPv4 address input entity.

IP to URL - The input for this transform is the IPv4 Address. This transform will try to find if there is a match between the IPv4 address input entity and a Target URLs by the WAS scans.[18]

Retrieve Scan Groups - The input for this transform is the IPv4 Address. This transform will search for every VM Scan Group where the IPv4 address input entity is or were a target.

Get Scan - The input for this transform is the Target IPv4 Address. This transform will retrieve the VM Scan associated with the Target IPv4 Address[19].

Get Scan Groups - The input for this transform is the Target IPv4 Address. This transform will retrieve all the VM Scan Groups where this IP address is part of its targets.

Get Vulnerable Services - The input for this transform is the Target IPv4 Address. This transform will retrieve all the Vulnerable Services reported in the VM Scan associated with this IPv4 Address.

Target to URL - The input for this transform is the Target IPv4 Address. This transform will try to find a match in the WAS scans' results where the Target URL resolves to the same IPv4 Address.

Get Vulnerability - The input for this transform is the VM Classification. This transform will retrieve the VM Vulnerability associated with it.

Get Scan Group - The input for this transform is the VM Scan. This transform will retrieve the VM Scan Group associated with this VM Scan.

Get Targets - The input for this transform is the VM Scan. This transform will retrieve all the Target IPv4 Addresses configured in this VM Scan.

Get All Targets - The input for this transform is the VM Scan Group. This transform will retrieve all the IPv4 Addresses that were ever associated with this VM Scan Group.

Get Scans - The input for this transform is the VM Scan Group. This transform will retrieve all the VM Scans that have occurred belonging to this VM Scan Group.

Get Classification - The input for this transform is the VM Vulnerability. This transform will retrieve the VM Classification associated with this VM Vulnerability.

[18]This is possible thanks to ElasticSearch index "scan_event", which in case of the WAS scans, saves in two properties the URL and the matching IP address.

[19]Entities can have properties in Maltego, and it is the case of Target IPv4 Address. This entity contains a property that stores the scan instance name.

Get Vulnerable Service - The input for this transform is the VM Vulnerability. This transform will retrieve the Vulnerable Service associated with this VM Vulnerability.

Get Vulnerable Targets - The input for this transform is the VM Vulnerability. This transform will retrieve all the Target IPv4 Addresses where this vulnerability was reported.

Get Information Gathered - The input for this transform is the Vulnerable Service. This transform will retrieve the Information Gathered associated with the Vulnerable Service. This transform usually is applied to a specific Vulnerable Service entity, and the entity is identifiable by the value it contains, which is "N/A".

Get Vulnerabilities - The input for this transform is the Vulnerable Service. This transform will retrieve all the VM Vulnerabilities associated with this Vulnerable Service according to the VM Scan that it is associated.

Retrieve Vulnerable Targets - The input for this transform is the Vulnerable Service. This transform will retrieve all the Targets IPv4 Addresses where this service had been reported.

Image 5.47 will illustrate the developed transforms that apply to the WAS scans.

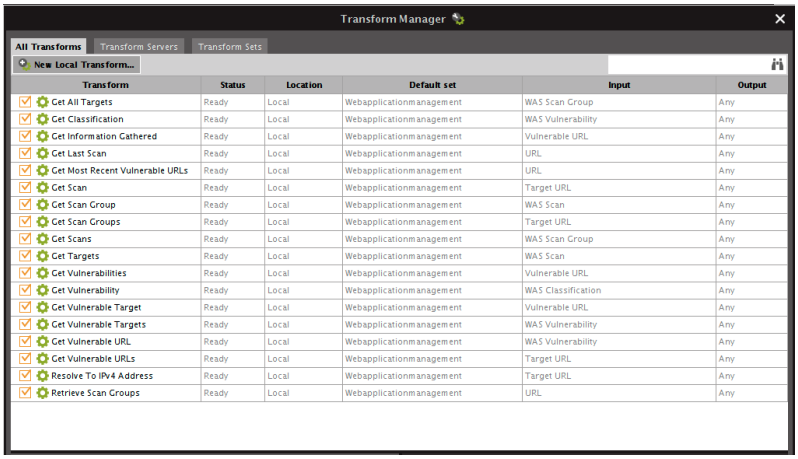
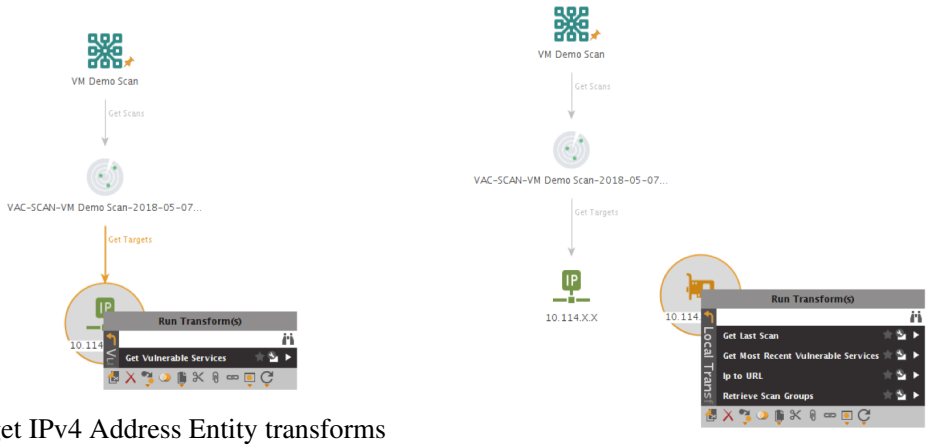


Figure 5.47: Maltego transforms concerning WAS Scans.

Given the similarity between the ones illustrated in image 5.47 and the previous just described, these will not be further analyzed.

Images 5.48a and 5.48b are an example of how the transforms are presented, and that different transforms appear to different entities.

Image 5.49 presents an example of the Maltego, with a graph generated from VACv2's data.



(a) Target IPv4 Address Entity transforms

(b) IPv4 Address Entity transforms

Figure 5.48: Example of different transforms for different entities.

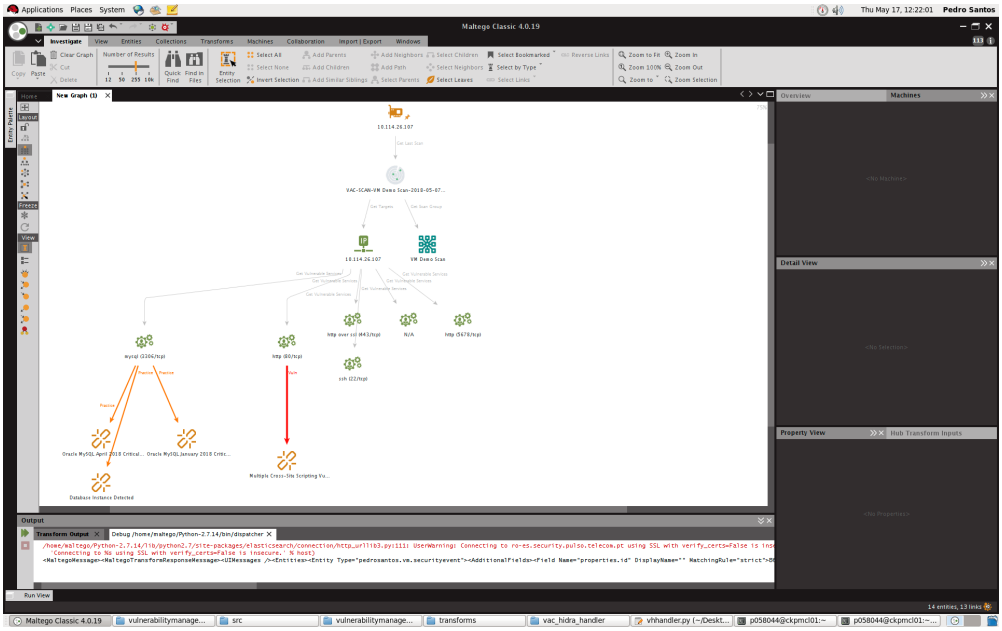


Figure 5.49: Maltego Graph Demonstration.

In illustration 5.49, it is possible to see that the graph was originated in the IPv4 Address entity where it was applied the Get Last Scan transform. The result generated an association between the input entity and the result, a VM Scan entity. The description next to the grey arrows declare the transform that has been applied, and the point of the arrows present the result of that transform. Except for the orange and red arrows, which symbolize a potential vulnerability or a confirmed vulnerability, respectively, these associations are the result of the Get Vulnerabilities transform.

5.3 Conclusion

In this chapter, the reader has just witnessed the final architecture of VACv2. This master thesis was divided into three implementation phases to match the objectives, and which purpose was to improve the in-house software named VAC.

The first goal was the overall improvement of the software, and it was without any doubt the most crucial of three objectives, this one was what allowed a smooth implementation of the remaining goals. This goal restructured VAC completely, it set straight the purpose of each block in the solution. It was thought to ease the addition of new technologies, and to provide VACv2's operator with more usability and power of the tool. One essential achievement was the layer of protection added to VACv2. From the secure communication between client and server to the authentication of the User against PT's Active Directory through Kerberos Authentication protocol. From the users allowed to access the software, to the cookie-based session. All these enforce security to the solution, which handles sensitive information, *i.e.*, vulnerabilities belonging to essential targets.

The second goal was much smoother considering the whole re-design of VAC into VACv2. The solution becoming unattached from any scanning technologies, made all the necessary development be the creation of a new class which had to extend the Appliance-Instance class, and that was it.

However, as the scanning technology did not offer a ruby library, a custom had to be developed. This second goal consisted of the addition of the Qualys Cloud-Based technology into the solution, but this technology contained some special perks when comparing with the others. The first is the fact that it is a Security as a Service (SECaaS) and PT does not manage the solution directly, something that did not occur with the other technologies and VACv2 had to be capable of handling it. The other feature was the fact that this technology contained a Web Application Scanning module, something that VAC was not capable of handling. However, like said before, the first goal of this master thesis was supposed to allow a more natural addition of new technologies into the system, and also to contemplate the possibility of having more than one scan type, easing the implementation of the second phase.

The third and last goal consisted of a full restructure of the Results Module, which in VAC due to the lack of time, was set aside, not providing any control to the operator about the data to be uploaded or the systems in which VAC was integrating. The result was what was developed in VACv2, turning this module pretty similar to the Scanner Manager module with the concern of easing the upload of data into new technologies. While also providing control over the platforms in which VACv2 will be integrating with, and allowing more natural vulnerability management concerning the exception and exclusion of events.

The redevelopment of this module had another feature which was a specialized development for the Maltego technology, which is a correlation data software. PT had full

interest in including the results provided by VACv2 on this platform, which will provide more information to the SOC team which are the typical users of this software.

The end of this additional development also ended the development phase of this project.

Chapter 6

Results

In the past two chapters, we have analyzed the objective, design, and development for the Continuous Security Assessment project. More specifically, chapter four depicted what the objectives were for the project. While chapter five described VAC software, as it was at the beginning of this project, and then described the new architecture, the ideas behind every implementation, the changes made, and so on.

VAC was the result of a thesis previous to this one. However, the software never made it to a production environment. The main reason was that PT did not renew its contract with Rapid7 concerning Nexpose. In other words, there is no real base of comparison on how to evaluate VACv2.

This thesis comes to release DCY's personnel from having to concern with vulnerability scans, and to allow a more natural data correlation of the vulnerabilities found in the assessed platforms, which otherwise would consist of intensive labors. This project through the use of VACv2 was made to help in the prevention and mitigation of possible malicious attacks. While also contributing with its results to a DCY more aware of the current problems, and allowing a more accurate and faster data correlation. It was suggested to present a quiz in this section of the report, but as there was only one person in charge of the vulnerability scans in PT, it would not bring that much value into the report.

This chapter illustrates, two recurrent procedures carried out by the DCY operator when handling scanning technologies, and these procedures will be illustrated for both Nexpose as Qualys. Afterward, we will evaluate the corresponding procedures in VACv2 software and verify potential gains *vs.* drawbacks. The procedures being analyzed will be the scan configuration and the report configuration.

6.1 Nexpose

6.1.1 Scan Configuration: Required Steps

Let's now analyze the necessary steps for setting up a scan in Rapid7's Nexpose.

For a scan to be set up, first the assets have to be configured in the platform. However,

the assets have to be arranged by network first. Nexpose contains an object, named Site, and this objective is responsible for aggregating the assets.

As PT contains multiple subnetworks, if the arrangement by network was not made, then only the assets configured in the same network as the appliance responsible for performing the scan would return results, and all the others would be wrongly considered as dead hosts. In worse case scenario there will be as many Sites as internal networks, meaning this procedure would have to be executed multiple times.

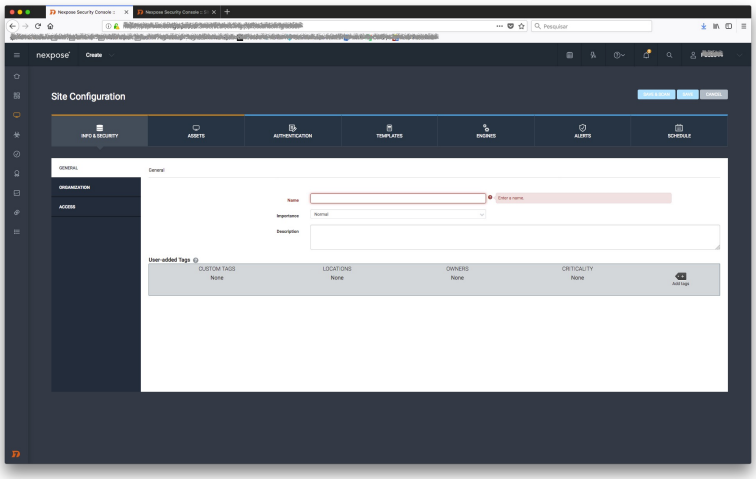


Figure 6.1: Nexpose Scan Configuration Step 1 - Identifier of the Site Object.

Screen 6.1 is illustrating the first page when configuring a Site object in Nexpose, and it will store the information related to a set of assets. The Name property is what identifies the Site.

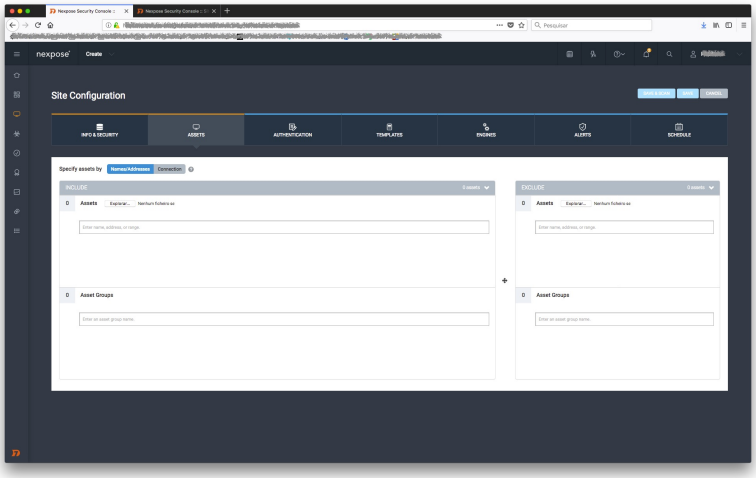


Figure 6.2: Nexpose Scan Configuration Step 2 - Declaring the IP Addresses of the targets of the scan.

Screen 6.2 is illustrating the page where the assets are to be configured.

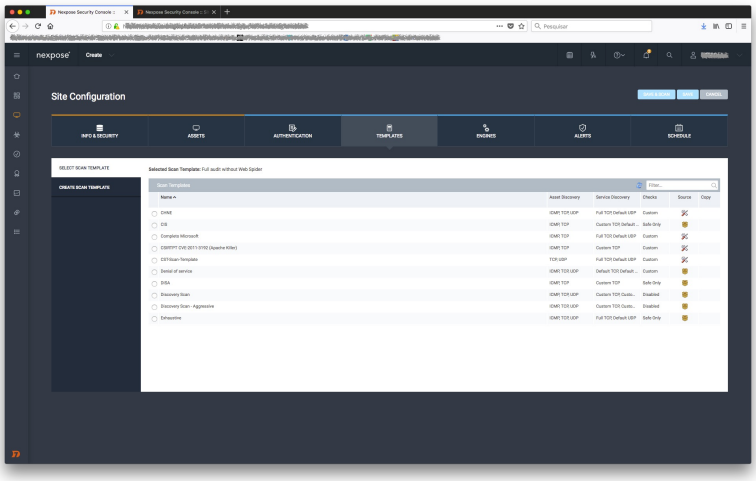


Figure 6.3: Nexpose Scan Configuration Step 3 - Choosing the scan template.

Screen 6.3 is illustrating the page where the template is chosen.

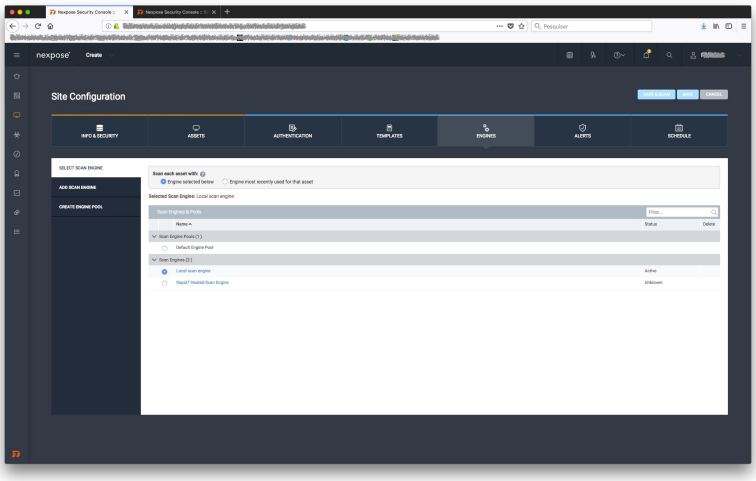


Figure 6.4: Nexpose Scan Configuration Step 4 - Associating the Site object with an Appliance.

Screen 6.4 is illustrating the page where the association between the assets and the appliance to perform the scan is done, thence the assets being group by IP address, because the ones contained in this Site have to be in the same internal network as the appliance chosen. At this point, the site will be assigned to an appliance.

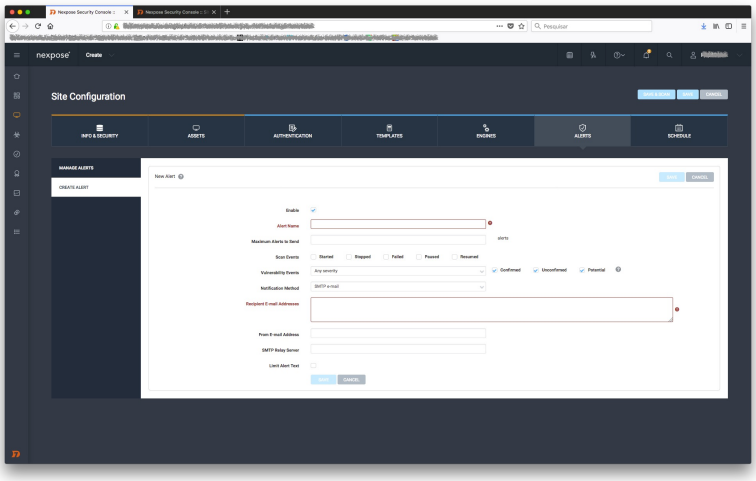


Figure 6.5: Nexpose Scan Configuration Step 5 - Enabling notifications.

Screen 6.5 is illustrating the page where the notifications are configured. The notifications might be sent accordingly to the actual scan state, or the type of event discovered.

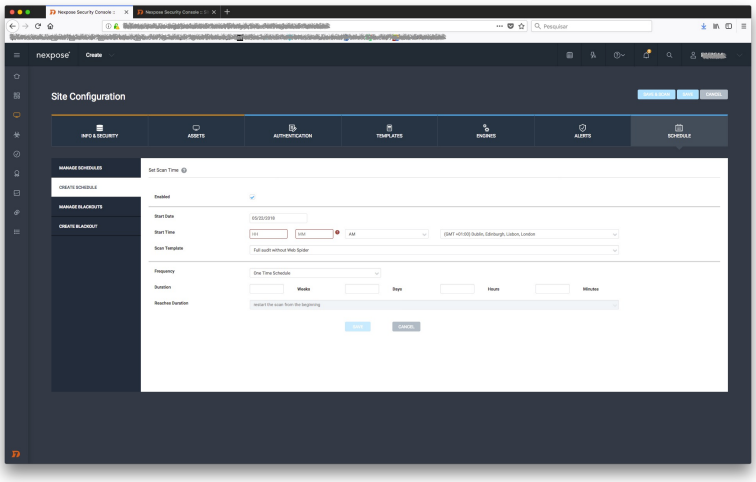


Figure 6.6: Nexpose Scan Configuration Step 6 - Configuring the scan recurrence.

Screen 6.6 is illustrating the page where the recurrence of the scan is determined.

A typical platform belonging by PT contains assets in multiple networks, which means that there has to be the need to create as many Sites as the number of different networks in that platform. The problem lies in the next step, Nexpose does not allow to aggregated this Site objects, meaning all the assets will run a different scan, in other words, if the scanning schedule ever needed to be changed then every Site would also have to be altered.

6.1.2 Report Generation: Required Steps

After a scan occurrence, the DCY operator needs to extract the results from the platform, and this is done through the generation of a report, which will be illustrated next.

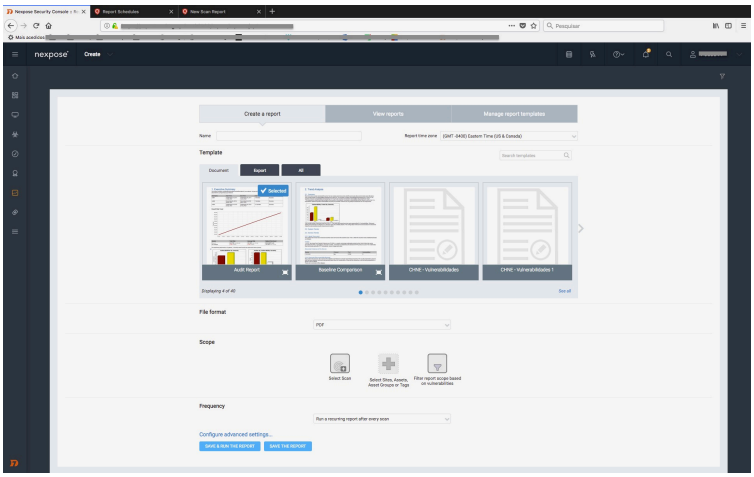


Figure 6.7: Nexpose Report Configuration.

Screen 6.7 is illustrating where the report generation is configured. We will now analyze the properties associated with the report configuration.

- Template property is the blueprint for the report to be generated, in other words, it will define what types of vulnerabilities should be presented, or which graphics to show in the report, among a lot of other options.

- File property will define the type of report to be generated. DCY always works with two types, a CSV file, and a PDF file.

- Scope property represents what should be the focus of the report, *i.e.*, scans or Sites. The difference, between these, a scan report will focus on the difference from the previous scan to the actual one, while a site report will focus on the state of the assets in the previous scan *vs.* the actual one.

- Frequency property, which turned out pretty handy because as the illustration shows, it is possible to set the frequency of the report for whenever the associated Scan or Site is scanned.

As DCY operates with two file types of the same report - CSV and PDF -, this means that this procedure has to be set up twice. However, there is no need of coming back to these configurations, unless it is intended to make a significant change to the report, *e.g.*, set a new template or file format. Even if the scan is canceled, thanks to the frequency property there is no need to come back to the report configuration.

6.2 Qualys

6.2.1 Scan Configuration: Required Steps

Qualys's scan configuration is pretty similar to the Nexpose's one. However, it has a few teaks. When setting up a scan, the IP addresses have to be previously stated into the Qualys platform. Otherwise, Qualys's platform refuses to scan the asset.

Qualys aggregates assets, which have to be arranged by a network, the object is named AssetGroup but contains a similar purpose to Nexpose's Site.

In Qualys, there has to exist as many AssetGroup as there are networks. The AssetGroup can be assigned to a given appliance, and that appliance will be the one to perform the scan. Qualys contains a Scan object which will contain the scan configuration. Is part of the scan configuration, to state the AssetGroup that have to be scanned, meaning that, in reverse to what happened in Nexpose, Qualys only needs to launch one scan and all the associated AssetGroups will be scanned despite their networks, thanks to the AssetGroups being associated with its correspondent appliance.

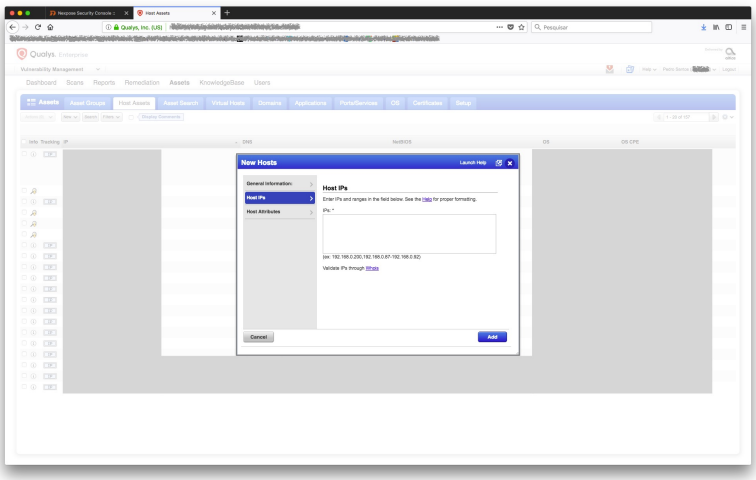


Figure 6.8: Qualys Scan Configuration Step 1 - Declaration of the targeted IP Addresses in the platform.

Screen 6.8 is illustrating the page where the target's addresses have to be stated. Otherwise, Qualys refuses to perform the scan.

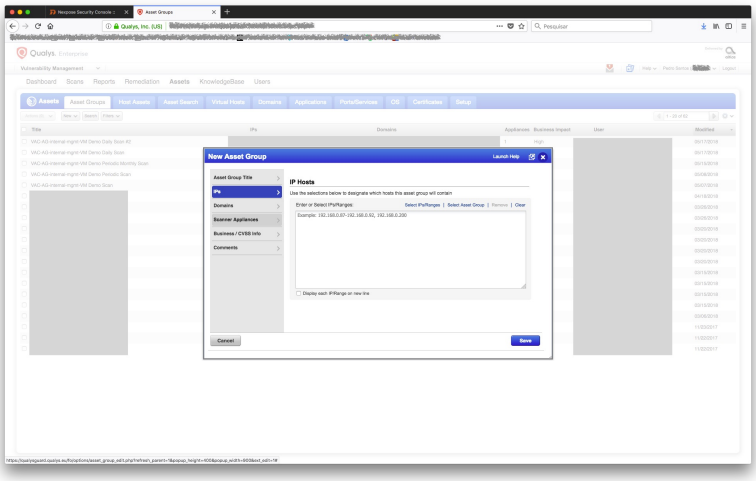


Figure 6.9: Qualys Scan Configuration Step 2 - Creation of an AssetGroup, which will aggregate the targeted IP Addresses to a scanning appliance, both must be in the same network.

Screen 6.9 is illustrating the page where the AssetGroup object is created, the next tabs will not be illustrated, but they will be described.

Asset Group Title Tab where the name of the Asset Group is stated.

IPs Tab where the IPs related to this AssetGroup are stated.

Scanner Appliance Operator assigns an appliance to the AssetGroup. The appliance is responsible for scanning a given network, "demanding" the AssetGroup only to contain IP Addresses located in that network. Any IP Address contained in the AssetGroup that is not included in that network might be wrongly considered as a dead host.

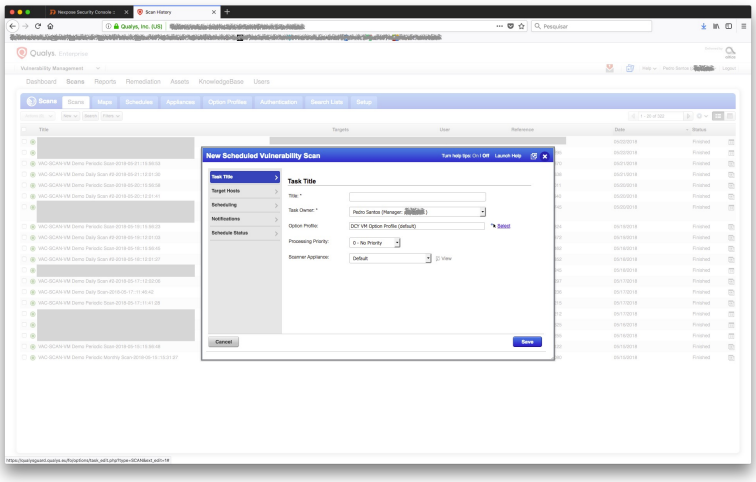


Figure 6.10: Qualys Scan Configuration Step 3 - Creation of a Scan Configuration.

Screen 6.10 is illustrating the first page when configuring a scan in Qualys. In this page, is set the title that identifies the scan configuration. Also, it is the place where the scan template is chosen, which in Qualys, the property is named as “Option Profile”. Another property is the “Scanner Appliance”, which will allow the operator to override the Scanner Appliance assigned to the AssetGroups using the defined in the property instead. To use the appliances already associated with the AssetGroups it only needs to set the value “Default”, as it was displayed.

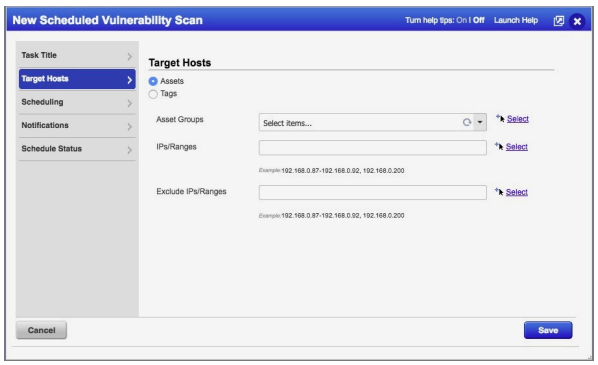


Figure 6.11: Qualys Scan Configuration Step 4 - Association of the Scan Configuration object with the AssetGroups.

Screen 6.11 is illustrating the place where the AssetGroups are to be defined.

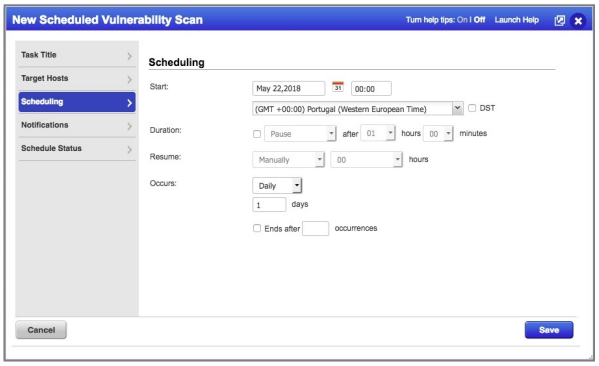


Figure 6.12: Qualys Scan Configuration Step 1 - Declaration of the Scan Configuration recurrence.

Screen 6.12 is illustrating the place where it is configured the scan recurrence.

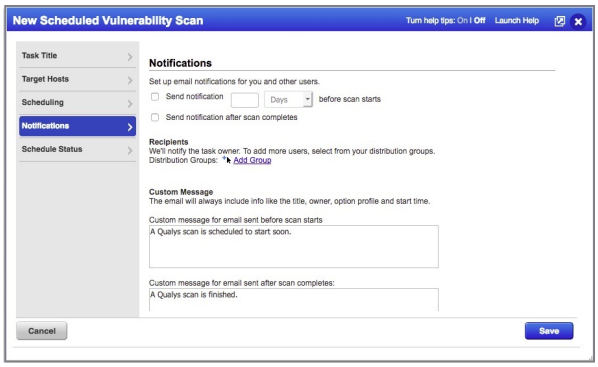


Figure 6.13: Qualys Scan Configuration Step 1 - Enabling notifications.

Screen 6.13 is illustrating the place where the notifications about the scan are configured. In Qualys, there is an object called Distribution Lists which aggregates emails, meaning that the emails contained in the selected Distribution Lists will get notified. Also, it is possible to customize the notifications to be sent.

From what we have seen, Qualys offers the possibility of performing one scan over all the associated Asset Groups or to performing a scan by Asset Group. This is left to the operator's choice.

The procedure illustrated above is an illustration of the Vulnerability Management scan type, the Web Application Scanning is similar. Therefore the WAS scan configuration will not be illustrated.

6.2.2 Report Generation: Required Steps

Figure 6.14 illustrates the page where the report generation is configured in Qualys. Like in Nexpose, here the "Report Template" property also represents the guidelines for the creation of the report.

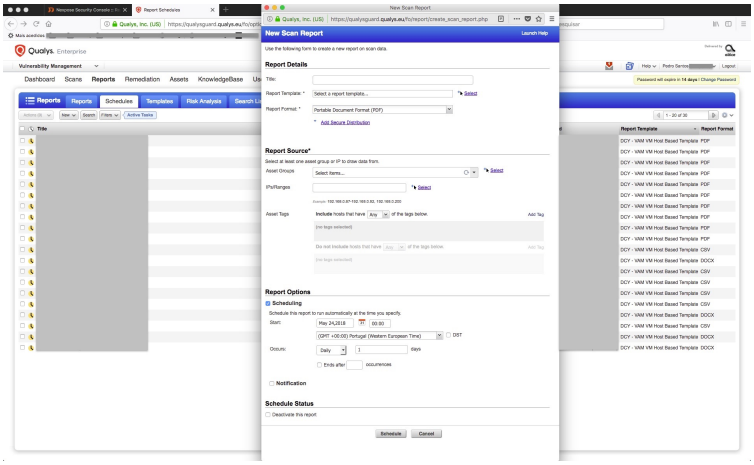


Figure 6.14: Qualys Report Configuration.

In screen 6.14, exists two differences when comparing it with Nexpose. The first is the Scheduling type, which is one of the most significant setbacks of Qualys concerning the effect on the work of the DCY's operator.

In Qualys, a Report has to be scheduled just like a scan, and can be not associated with the scan task, meaning that there is a need of being confident that the scan has already finished for obtaining the correct results. Otherwise, if the scan is still running and the report starts running, it will fetch the previous scan returning deprecated results.

The second difference is the Notification area, which although not displayed, in case of the property being checked it would behave as the scan notification. It will provide the operator with the power of choosing the Distribution Groups to be notified, and also the possibility of customizing the notification.

The remaining properties are pretty straightforward and self-explanatory. The title is the property which unequivocally identifies the report. The report format property is the format for the report to be generated. The Report Source is the target for the report. Qualys' reports contain the same problem as the Nexpose's ones, which is the need of having to configure two reports for a scan - one in PDF and the other in CSV.

The WAS Report contains the same structure as the VM report, meaning it will not be illustrated.

6.3 VACv2

6.3.1 Scan Configuration: Required Steps

Although already shown in previous chapters, let's now appreciate the simpleness associated with the configuration of a scan in VACv2, illustrating both scan types, VM and WAS.

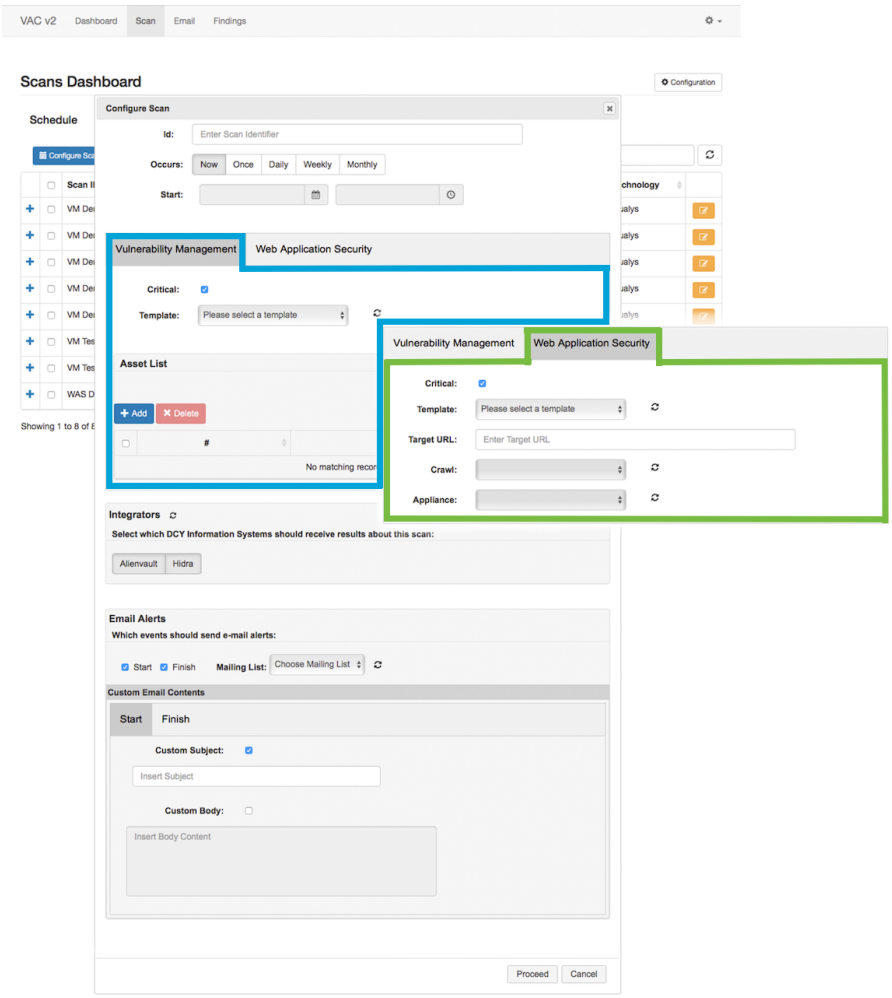


Figure 6.15: VACv2 Scan configuration page.

Screen 6.15 is illustrating the place where the scans are configured in VACv2 - VM and WAS, highlighted in blue and green, respectively. In this page, it is possible to pinpoint almost if not all the properties that were analyzed in the previous sections regarding Nexpose and Qualys.

At first glance, the user-friendly interface provided by VACv2 offers the user a more comfortable experience, and in general, makes VACv2 easier to understand and operate.

Let's analyze the fields displayed in illustration 6.15:

Id The name of the scan configuration.

Recurrence Defines the recurrence of the scan. Accordingly, to the chosen option, it will present more or fewer fields to fulfill.

Scan type Chosen by selecting the correspondent tab. Will present the properties associated with the scan type. Independently of the scan type, the user will always have to fill the critical and template options.

- When the user chooses the VM scan, highlighted in blue, it will have to fill in all the targets of the scan in the Asset List table.[1]
- If the user chooses the WAS scan, highlighted in green, when inserting the Target URL, VACv2 will automatically recommend an appliance to perform the scan, which will be presented to the operator in the "Appliance" field. In reverse to what happened in the VM scan, the operator can supersede this option by selecting the appliance himself. Besides these two fields, there is one more field that needs to be configured which is the "Crawl" option, that will define the length of the scan. This property will set the limits for the scan, *i.e.*, will define if the appliance should scan everything below the URL or a specific sub-domain.

Integrators This property is optional, but if defined, it will determine the DCY's information repositories where the results should be uploaded.

Email Alerts This last configuration is optional and can be divided into two sections. The first will only activate notifications regarding the scan, by determining which scan states should generate a notification and to which Mailing List should the notifications be sent. The "Custom Email Contents" section will allow the operator to customize the title or the message to be sent in the notifications that will be sent to the distribution list. Under the attachments area, the attachments B.1, and B.2 represent a notification of the starting of a scan, and the ending of a scan respectively.

6.3.2 Report Generation: Required Steps

In VACv2, there is no concern with the report generation related to the scan event. It is an action that can be automatically be triggered when the scan is finished. There are two possible behaviors concerning the reports, the first - considered to be more often- will occur when choosing one or more integrators while configuring the scan. In this case, the report will be triggered, and the results will be uploaded into the corresponding information repositories, but only after the results are treated by VACv2. The second alternative is when no integrator is selected, in this case, the Scan Manager module will not trigger the Results Manager module, not generating any report. The DCY operator is able to access the scanning platform and produce a report at any time. The scans that might occur under these conditions are considered one time only scans.

VACv2 will generate the two reports handled by DCY - CSV, and PDF. The objective of the CSV report is for VACv2 to be able to treat the results, and the PDF will be avail-

[1]The user has to insert all the targets without having to worry if the IP addresses were already declared in the scanning platform, or about their networks. VACv2 will automatically declare the IP addresses in the scanning platform, and group them by network, assigning them the corresponding appliance to perform the scan.

able for the user to download from the Hidra information repository, but only when this integrator gets checked.

6.4 Nexpose and Qualys *vs.* VACv2

Now, after examining the regular procedure performed by the DCY operator in Nexpose, Qualys, and VACv2, we are ready to analyze the differences between the different platforms. Regarding the configuration of a scan, it was possible to witness the increased difficulty of doing so in a scanning technology *vs.* configuring one in VACv2. From the insertion of the IP Addresses/ URL to the report configuration, but that was precisely what was intended from VACv2, to automize and optimize the vulnerability management procedures performed by DCY. VACv2 aimed to be a centralized platform for managing scans, and this goal has some strings attached. When configuring a scan in VACv2, is pretty easy to change the scanning technology in charge of the scan. To make this happen, all the operator needs to change is the combination of the critical and template properties, this because VACv2 takes control of the scheduling instead of the scanning technologies. In reverse, if a scan was configured in a scanning technology, and if there were the need to change the scan to a different scanning technology, the operator would have to go the platform where the scan was initially configured, delete that configuration, and then recreate the scan in the new scanning platform, which is a procedure that might occur with some frequency.

The reason why this procedure might be frequent is that purchased scanning technologies have licensing limits, and PT contains thousands of IP addresses which if all had to be configured in a paid platform the costs would be unbearable. Meaning the management of the "slots" in this type of platforms is sensitive and has to be managed carefully. The operator has to prioritize the assets and configured them in the platform he finds to be more fitted given its criticality. However, PT is a company in which new assets appear almost every day, meaning the reassessment is a regular procedure.

VACv2 is also responsible for managing the assets, in other words, VACv2 inserts the assets into the scanning technologies before a scan and removes them when the scan finishes, this takes the load out of DCY of this terribly laborious task, while also providing more efficient use of the paid technologies.

Another example, PT acquired Qualys and stopped using Nexpose, all the scans configured in Nexpose had to be reconfigured in Qualys. In VACv2, the required procedure was only editing the scan and changing the Template property to one belonging to Qualys.

6.5 Information Systems

There are three Information Systems available in VACv2's Results Manager module - AlienVault, ArcSight, and Hidra -, although only two are currently active, AlienVault and Hidra. ArcSight is not active because PT decided it that way. Maltego, previously shown, was for demonstration purposes only.

Maltego was already illustrated in section 5.2.3 with a real case, so it will not be referenced again. So in the following section, we will be analyzing AlienVault and Hidra.

6.5.1 Hidra

When VAC was built, it was ready to upload the scan results to Hidra. The events were to be uploaded with Nexpose's format, OpenVAS results would have to be changed to Nexpose's format for being uploaded into the platform. We have already seen the format of the events that are being uploaded from VACv2 into Hidra, which are technology independent. However, as it never got to the production environment, there is no base of comparison of the data sent by VAC *vs.* VACv2. Under the attachments area, it will be possible to observe four examples of events uploaded into Hidra, one for each index by scan type:

B.3 Illustrates one event contained in the "scan_event" index, which is referent to a VM scan type. The "action_result_attachment" was shrunken in order to minimize the illustration.

B.4 Illustrates one event contained in the "scan_vuln" index, which is referent to a VM scan type.

B.5 Illustrates one event contained in the "scan_event" index, which is referent to a WAS scan type. The "action_result_attachment" was shrunken in order to minimize the illustration.

B.6 Illustrates one event contained in the "scan_vuln" index, which is referent to a WAS scan type.

It is through the data available in this information system, and because it is aware of the structure of both indexes, that Maltego is able to fetch the data in this information repository, and to correlate it, presenting it then to the Maltego operator.

6.5.2 AlienVault

As previously said, VACv2's AlienVault integrator uses the syslog protocol with the CEF format to upload its values into AlienVault USM. Next, it will be exemplified an event uploaded into the SIEM.

```
Nov  3 14:50:58 openvas CEF:0|DCY|VAC|2|838669|VAC-SCAN-VM
Teste Scan #2-2017-11-03::14:50:45|6|source=Qualys|
scan_type=VM|vuln_title=SSL/TLS Server supports TLSv1.0|
vuln_id=38628|vuln_type=Vuln|vuln_target=XXX.XXX.XXX.XXX|
vuln_policy=Excluded|vuln_policy\_description=Exclusão
exemplo.|vuln_category=General remote services|
vuln_protocol=tcp|vuln_port=443|vuln_service=http over
ssl|vuln_cve=|vuln_os_cpe=|vuln_pci=1|vuln_exploit=””|
vuln_malware=””|vuln_refs=””|vuln_status=”Active”|
vuln_result=”TLSv1.0 is supported”|vuln_threat=”TLS is
capable of using a multitude of ciphers (algorithms) to
create the public and private key pairs. For example if
TLSv1.0 uses either the RC4 stream cipher, or a block
cipher in CBC mode. RC4 is known to have biases and the
block cipher in CBC mode is vulnerable to the POODLE
attack. TLSv1.0, if configured to use the same cipher
suites as SSLv3, includes a means by which a TLS
implementation can downgrade the connection to SSL v3.0,
thus weakening security. A POODLE-type attack could also
be launched directly at TLS without negotiating a
downgrade. This QID will be marked as a Fail for PCI as
of May 1st, 2017 in accordance with the new standards.
For existing implementations, Merchants will be able to
submit a PCI False Positive / Exception Request and
provide proof of their Risk Mitigation and Migration
Plan, which will result in a pass for PCI up until June
30th, 2018.”
```

From the previous example, it is possible to understand how a message is uploaded with the CEF format, previously analyzed. It might seem confusing, but this is because of the extension property of CEF's format, which is responsible for carrying the actual message. In the example, the content of the extension property is also split by the pipe symbol (|), which means that the extension field starts at the "source" property.

The events uploaded are strictly vulnerability events. In other words, the contents of the ElasticSearch's index "scan_event" will not be sent, because SIEMs only care about events itself and not other pieces of information.

6.6 Conclusion

VACv2 has no base of comparison to any other software because VAC was not deployed in a production environment, and also because VACv2 came to mitigate a problem for PT which no other software attempts to, the management of assets/scans cross multiple scanning platforms. No quiz was made because only one person handled the vulnerability assessment routines in DCY. However, it is possible to compare procedures between software. The procedures that DCY personnel had to perform and that VACv2 come to relieve. Also, the new vulnerability's data available in the DCY's information repositories, which is now possible to correlate with other sources of information.

This chapter demonstrated the complexity of performing two procedures in the scanning platforms - scan configuration and report configuration - *vs.* making the equivalent procedures in VACv2. It was illustrated the simplicity behind VACv2's solution, and although the scanning technologies offered more options, VACv2 provided the standard features used by DCY.

Besides these points which were the primary focus of this project - the optimization of periodical scannings and a central point of scan management to relieve DCY personnel - this project also improved the results produced. VAC used the output format provided by Nexpose to upload its values into ArcSight and Hidra. As in VACv2, this structure has been revisited, by changing the original structure outputted - which was only ready to handle VM results - it is now possible to send values to more systems, which is the case of AlienVault. Also, the fact that information is getting to Maltego for correlation analysis -a brand new feature -, which will allow a more natural correlation of the data produced by the scanning technologies and other information repositories.

Chapter 7

Conclusion & Future Work

Given the urge for companies to protect themselves from potential attacks that might come to incur, the run for discovering vulnerabilities in their perimeter and inside is a must. To achieve this goal, there are a diversity of companies which their focus is to offer tools specialized in vulnerability management.

PT is one of the biggest telecommunications corporations in Portugal, for that reason it needs a good cybersecurity team with robust procedures for preventing potential attacks, and for that reason acquired vulnerability management tools. However, the procedures associated with the configuration of such tools, and given the size of PT, it would turn into an unbearable task of maintaining the assets updated in such tools. The limited time of the DCY - PT's Direction of CyberSecurity and Privacy - which are the ones responsible for performing these operations of vulnerability discovery and management, would suffer a considerable impact. The alternative to reducing DCY's required time in these operations while also improving the efficiency of the tools was to build a software. That software would become a central point for coordination and management of scans to PT, and it was named Vulnerability Assessment Coordinator - VAC.

VAC was the final product of a master thesis. It was designed to work in a specific scenario with specific tools. However, two factors sent VAC back to the shelve. One was the handling of the tool. It was not user-friendly and did not have the capacity for dealing with errors, which as it turned out demanding even more time from DCY than the original vulnerability management tools. The second factor was the scenario for which VAC was designed getting changed. These two circumstances determined the depreciation of the tool.

Nonetheless, PT did not forfeit of this project, for two reasons, the fact that they had spent financial and technical resources on VAC, and the second reason was because DCY continued to have a problem between hands, which was the necessary time for performing vulnerability management operations, these two circumstances lead to a new master thesis. VAC had potential, and PT decided to do a new master thesis - the Continuous Security Assessment thesis -, which would consist in an improvement over VAC, while also

expanding its scope to endure despite the technologies involved or even if the scenario might get changed.

VACv2 - which is VAC version two - development consisted in improving the usability and the security of the tool, while also making it unattached to any scanning technology, scanning type or results' platform. This master thesis would even reuse the data of the scans uploaded into Hidra - an information repository belonging to PT-, and made a specialized correlation data software - Maltego- download that information for improving the work of other teams like the SOC team.

There are multiple ways for this tool to keep evolving in the future. VACv2 handles the scan configurations as being the aggregators of the platforms, could be possibly easier to manage if there was an actual Asset Group module, where the assets would be configured, and the scans would have an association to them. Another possible work for the future is related to the configuration of a scan. Now, after understanding the complexity of performing vulnerability scans, it is easy to understand the importance of performing authenticated scans, these types of scans improve the level of confidence on the vulnerabilities reported by the scanning tools. The work would be about developing an extension in the scan configuration to allow the insertion of credentials associated with the assets. One last thing that comes to mind is the possibility of allowing the scans reports to be generated at all times by the scanning technologies - and not only when an integrator is chosen -, and then stored in VACv2.

Appendix A

VAC Interface Module

A.0.1 Scheduler Related

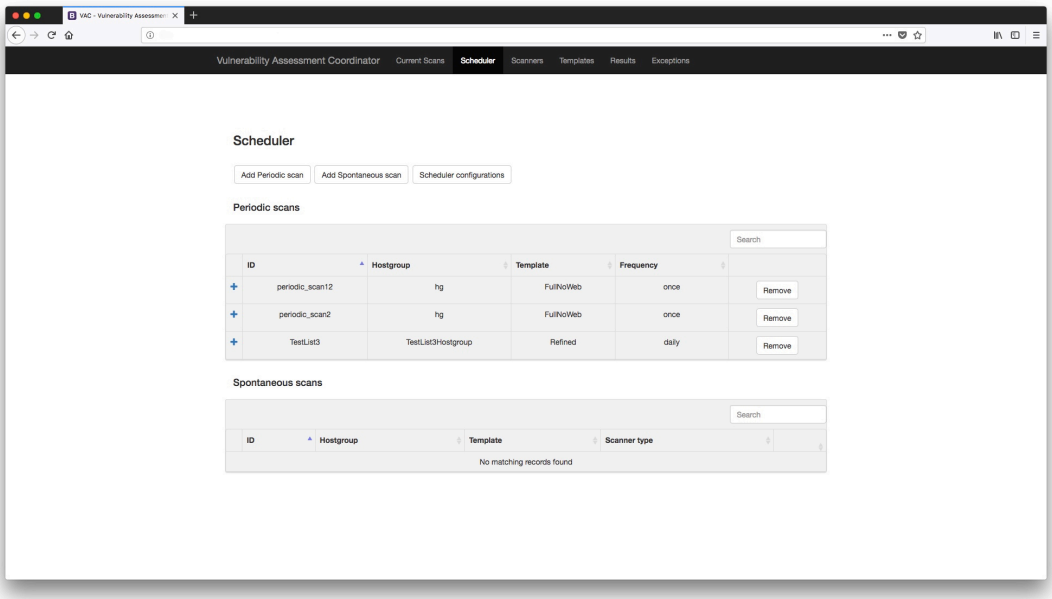


Figure A.1: Scheduler View

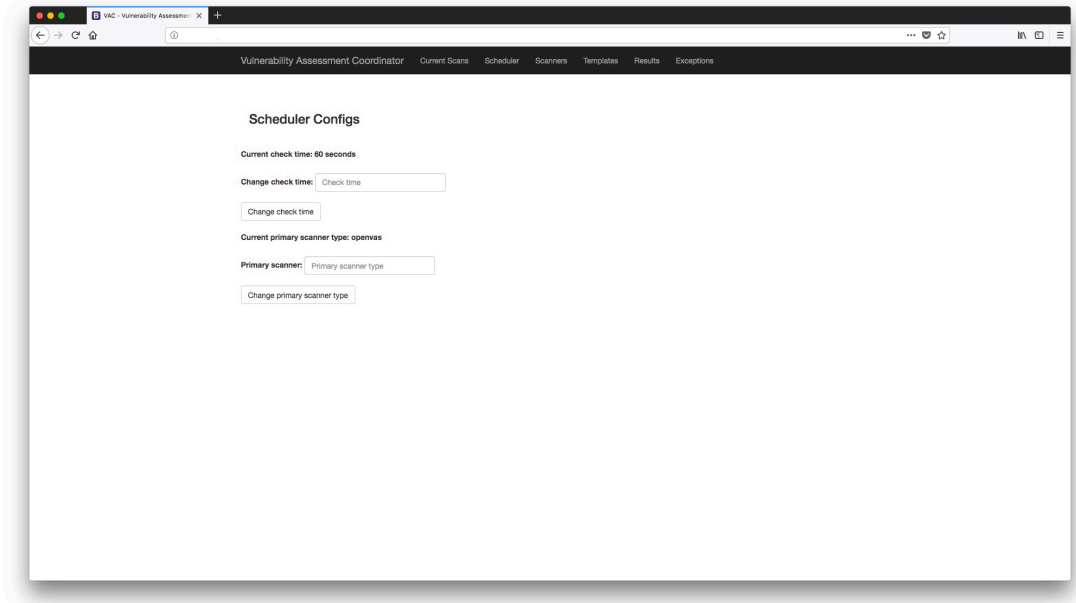


Figure A.2: Scheduler Configuration's Properties

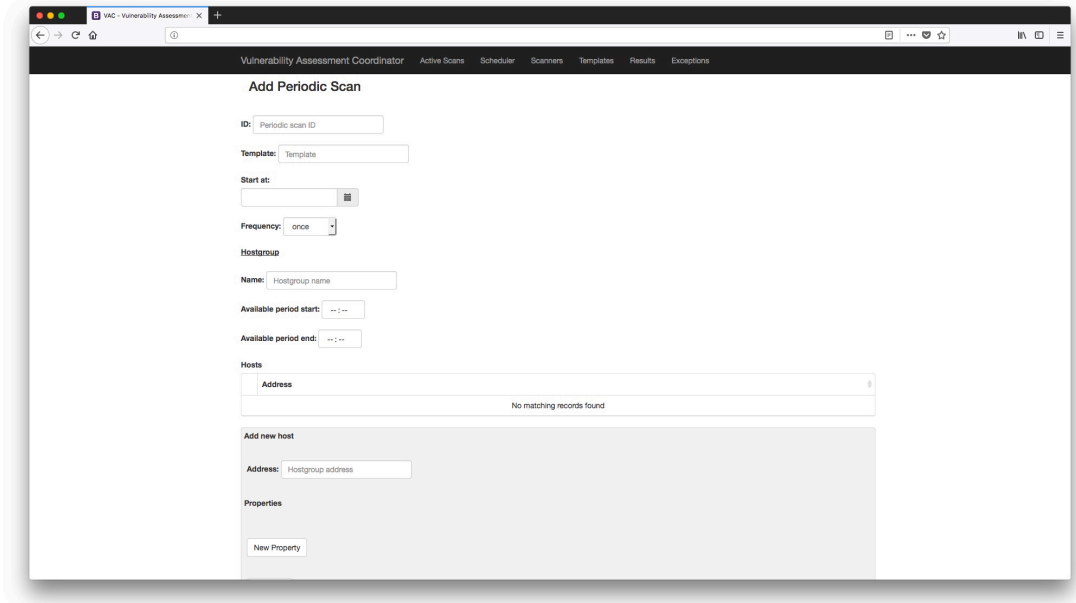


Figure A.3: Configure New Periodic Scan

A.0.2 Scanner Related

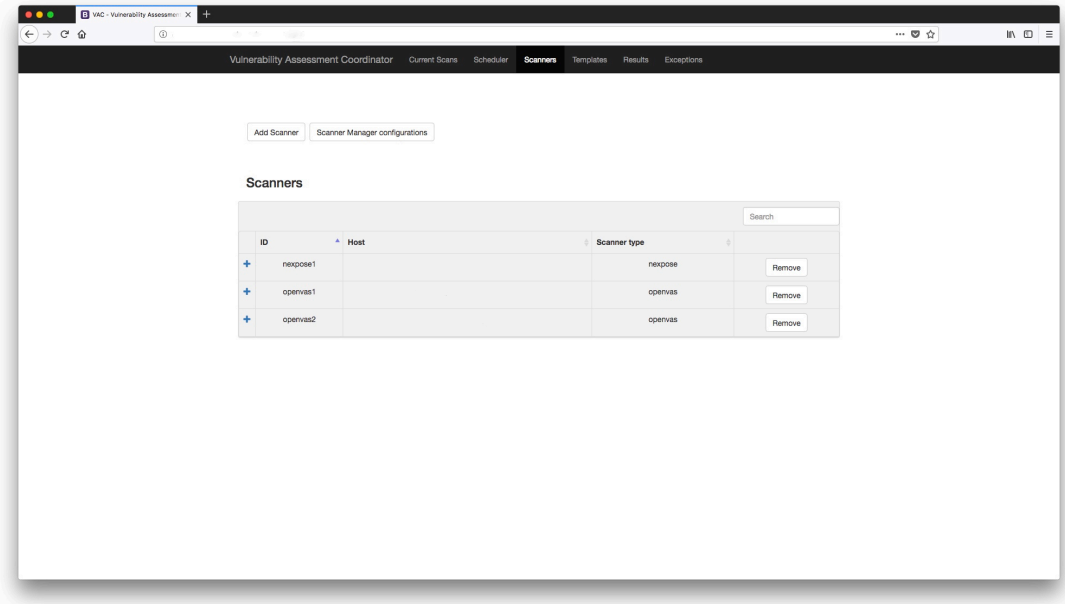


Figure A.4: Scanner View

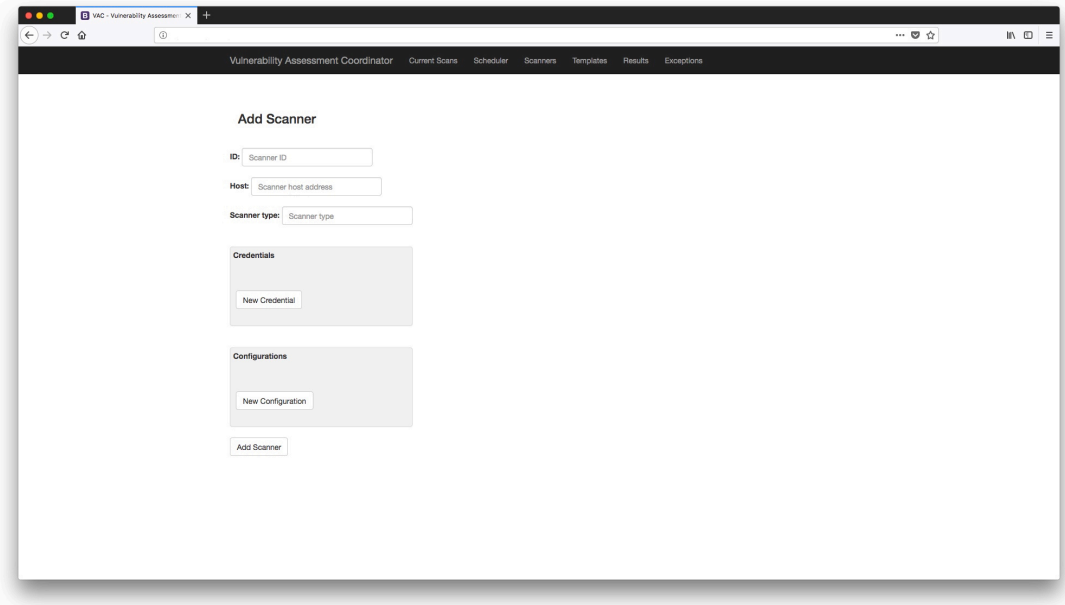


Figure A.5: Add Scanner Configuration

A.0.3 Results Related

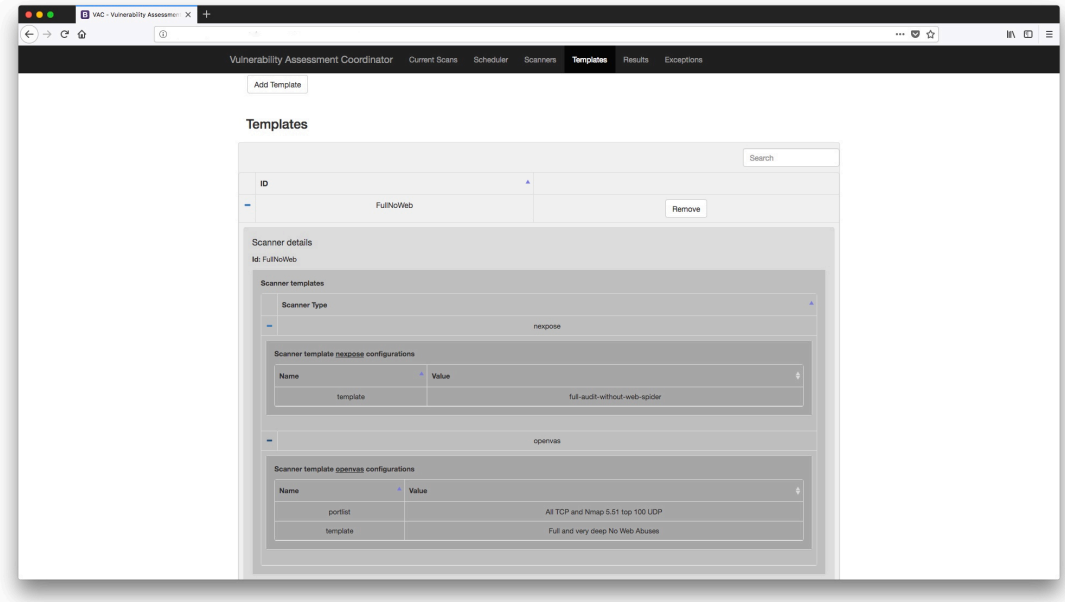


Figure A.6: Template View (Expanded)

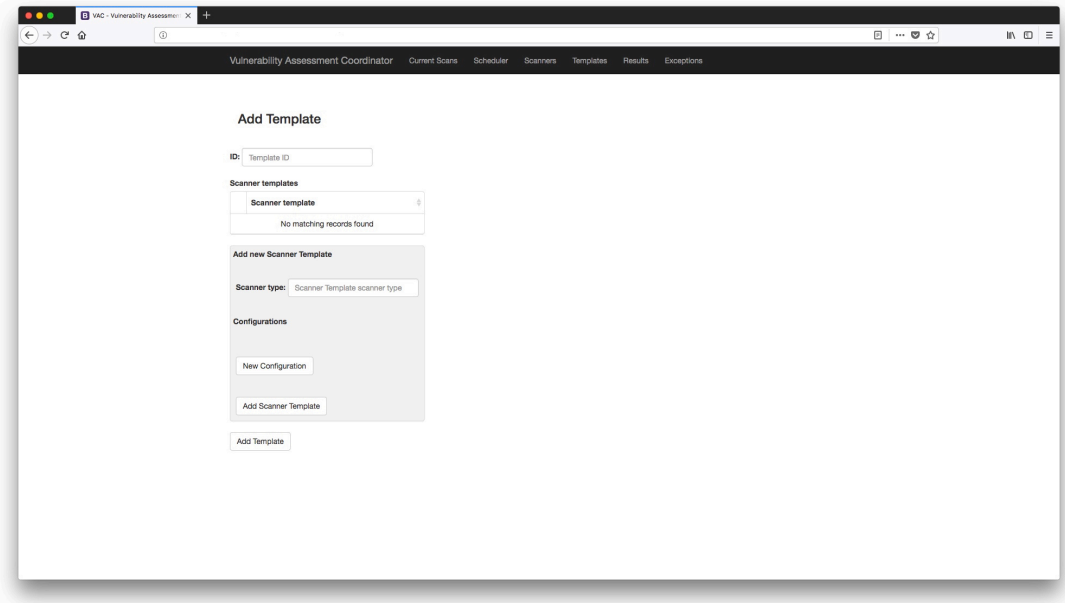


Figure A.7: Configure New Template

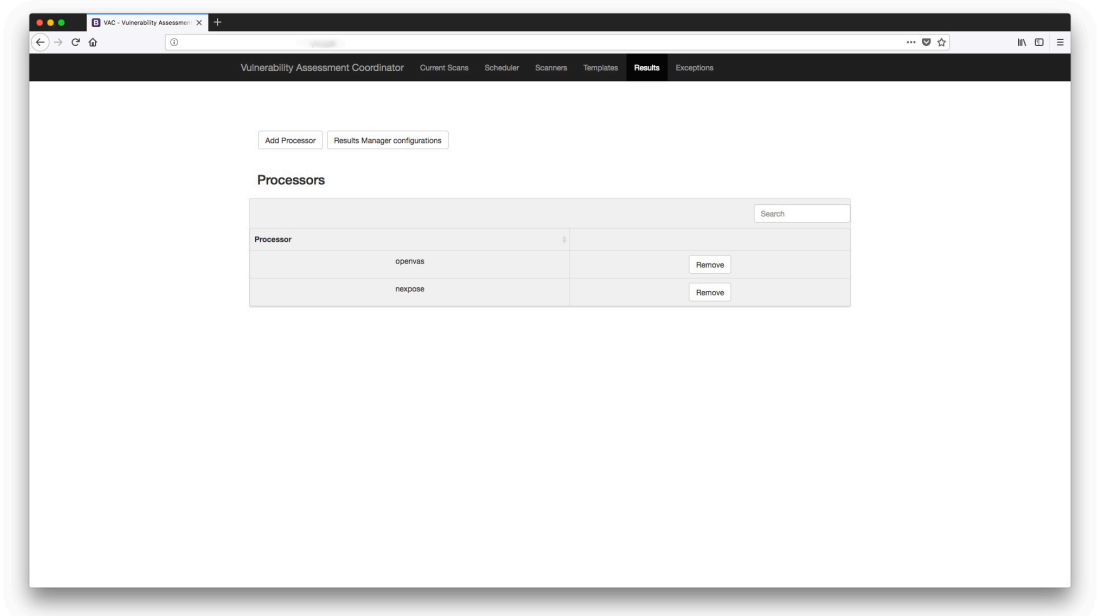


Figure A.8: Processor View

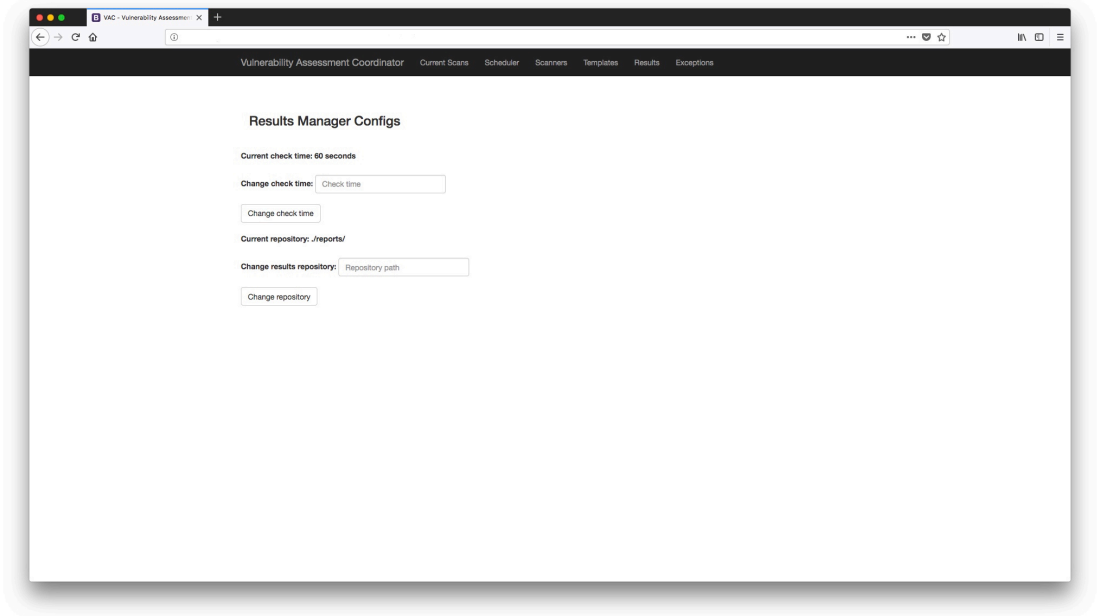


Figure A.9: Results Manager Configuration's Properties

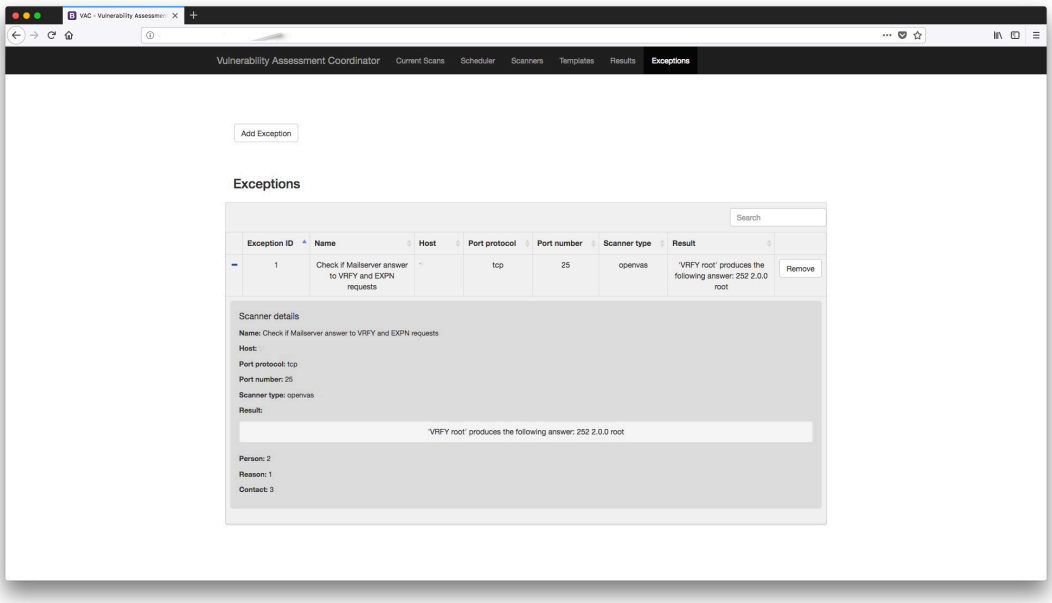


Figure A.10: Exception View (Expanded)

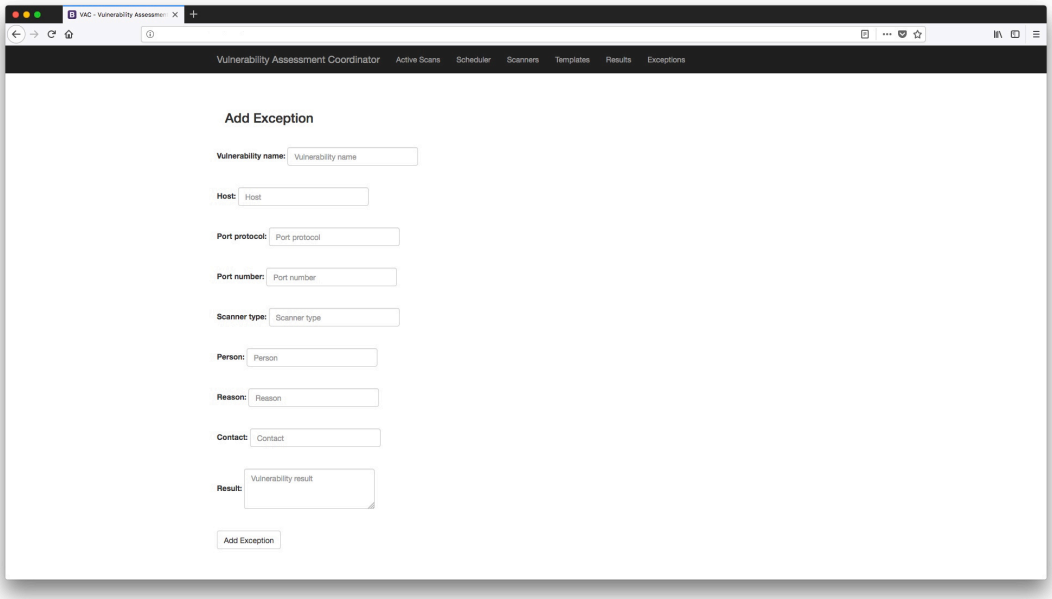


Figure A.11: Configure New Exception

Appendix B

VACv2

B.0.1 Email Notifications Examplified

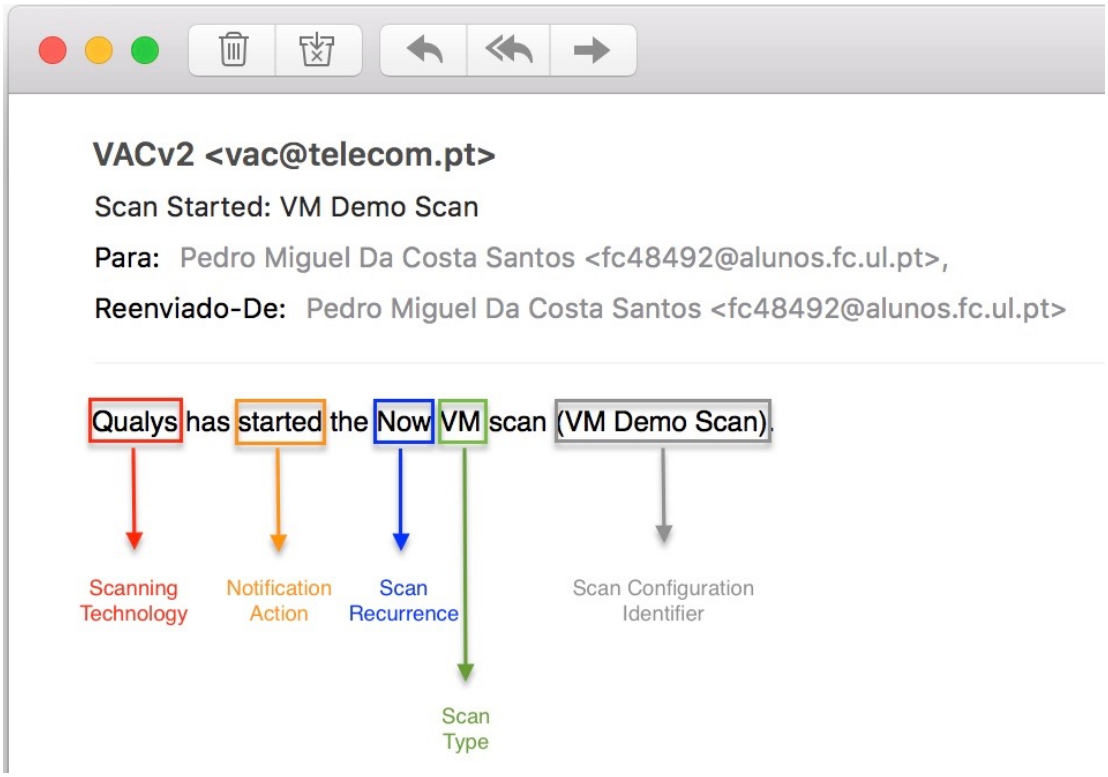


Figure B.1: Start scan action notification example.

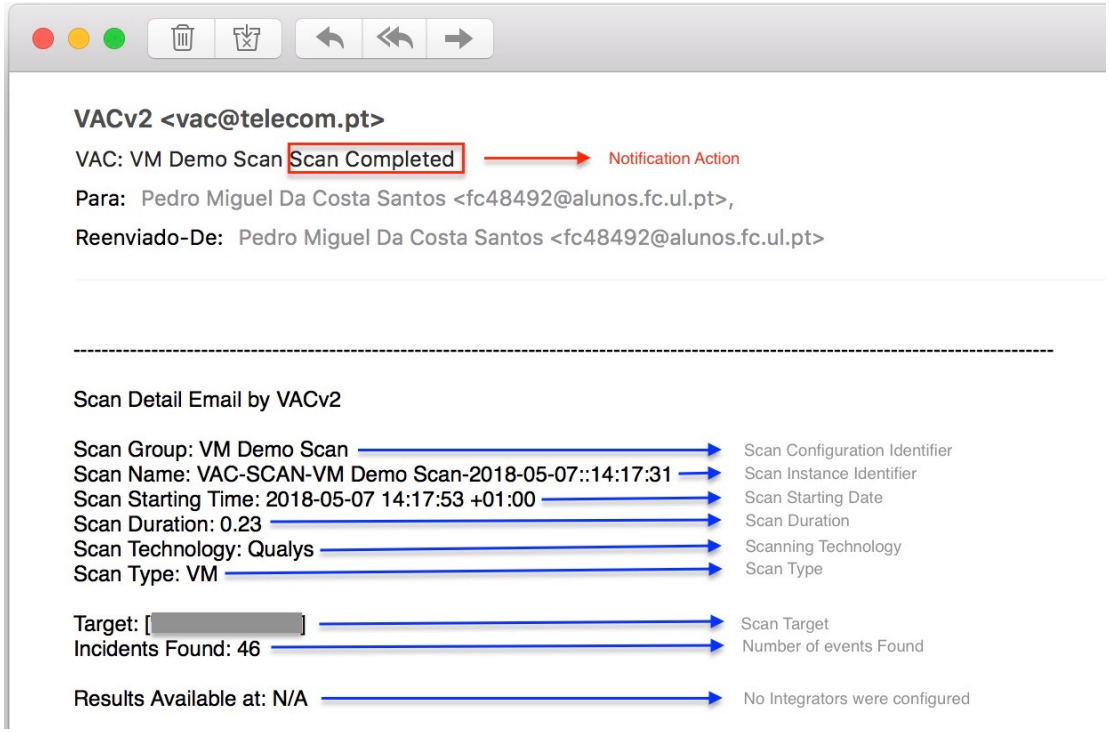


Figure B.2: End scan action notification example.

B.0.2 Hidra Information Repository structure

B.0.3 VM Scan Type Examples

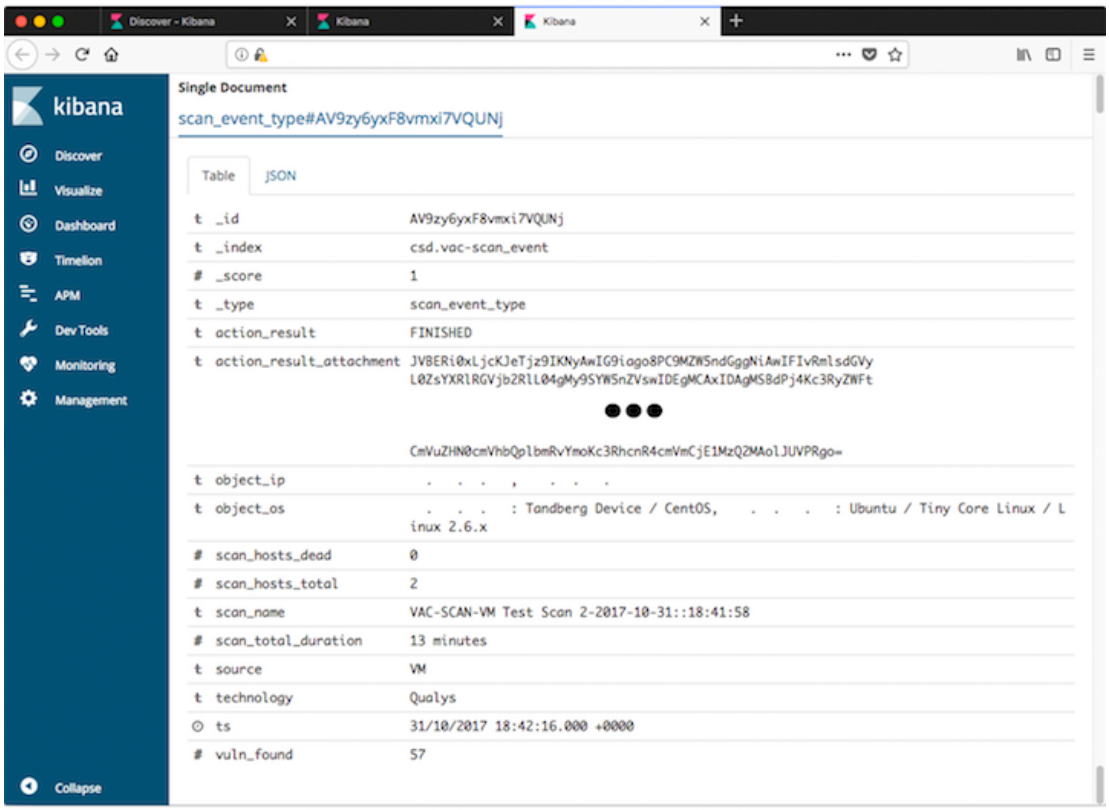


Figure B.3: ElasticSearch's scan_event example for VM scan type

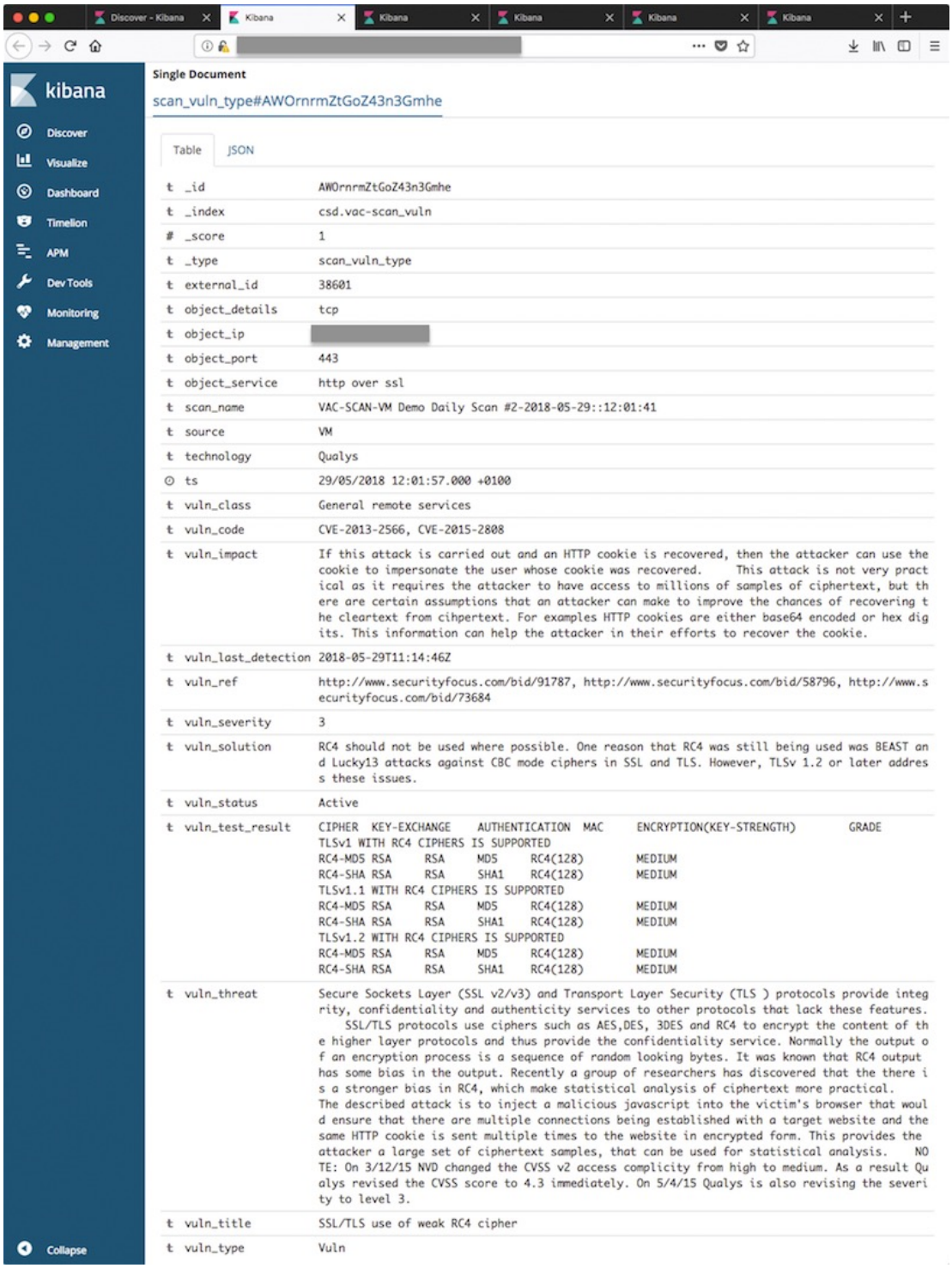


Figure B.4: ElasticSearch's scan_vuln example for VM scan type

B.0.4 WAS Scan Type Examples

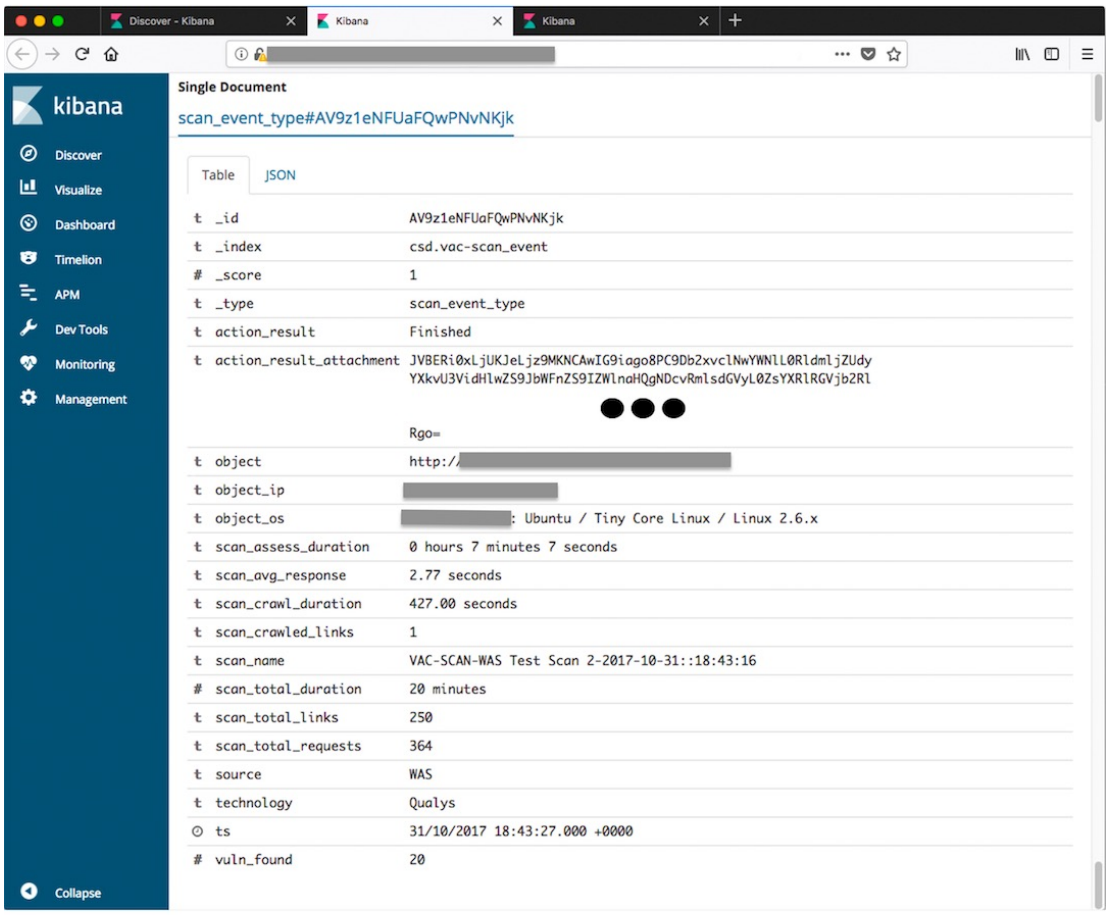


Figure B.5: ElasticSearch's scan_event example for WAS scan type

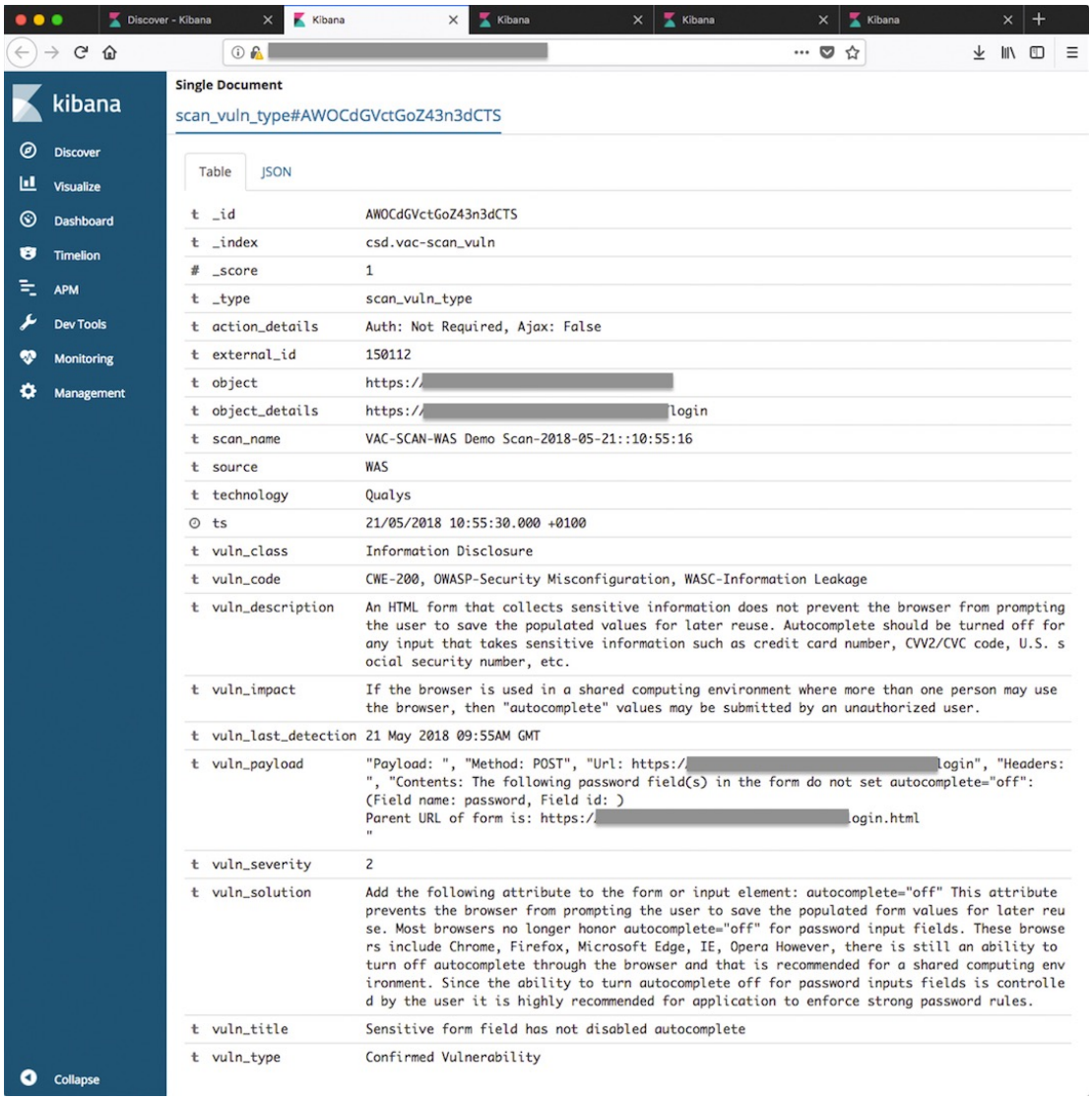


Figure B.6: ElasticSearch's scan_vuln example for WAS scan type

Abreviaturas

CIA Confidentiality, Integrity, Availability. 8, 16

CVE Common Vulnerabilities and Exposures. 7

CVSS Common Vulnerability Scoring System. 15

DCY Direção de Cyber Security and Privacy. vii

FCUL Faculdade de Ciências da Universidade de Lisboa. vii

FFCUL Fundação da Faculdade de Ciências da Universidade de Lisboa F.P.. vii

MSI Mestrado em Segurança Informática. vii

PT Portugal Telecom. vii, 16

SOC Security Operations Center. ix

VAC Vulnerability Assessment Coordinator. viii

Bibliography

[1] Yaml. https://pt.wikipedia.org/wiki/YAML, November 2016.

[2] Bootstrap. https://getbootstrap.com, December 2017.

[3] CERT. https://www.cert.org/vulnerability-analysis/, November 2017.

[4] Common vulnerabilities and exposures - vulnerability terminology. https://cve.mitre.org/about/terminology.html, November 2017.

[5] CVSS. https://www.first.org/cvss/v1/guide#2-0-Scoring, November 2017.

[6] ENISA - vulnerability definition. https://www.enisa.europa.eu/topics/threat-risk-management/risk-management/current-risk/risk-management-inventory/glossary, November 2017.

[7] IETF - RFC 4949. https://tools.ietf.org/html/rfc4949, November 2017.

[8] Impact of vulnerability. https://www.intel.com/content/www/us/en/security-center/impact-of-vulnerability.html, April 2017.

[9] Introducing json. https://www.json.org, November 2017.

[10] Uc davis. http://seclab.cs.ucdavis.edu, December 2017.

[11] Vulnerability definition. https://cve.mitre.org/about/, November 2017.

[12] Eric Alata, João Antunes, Mohamed Kaaniche, Nuno Neves, Paulo Veríssimo, and Vincent Nicomette. Critical utility infrastructural resilience project acronym: Crutial. pages 2 – 5, January 2017.

[13] AlienVault. Unified Security Management. https://www.alienvault.com/products, January 2018.

[14] TechRepublic Chad Perrin. Cia triad. http://www.techrepublic.com/blog/it-security/the-cia-triad/, December 2016.

[15] Cisco - Cisco Security Research & Operations. What is the difference: Viruses, worms, trojans, and bots? https://www.cisco.com/c/en/us/about/security-center/virus-differences.html, December 2017.

[16] The MITRE Corporation. Cve. https://cve.mitre.org, December 2016.

[17] The MITRE Corporation. Cve details. https://www.cvedetails.com/index.php, December 2016.

[18] M. Correia, N. Neves, and P. Veríssimo. Architecture of maftia. In *The Middleware Architecture of MAFTIA: A Blueprint*, December 2016.

[19] M. Correia and P. Sousa. Information security in software. In FCA editora, editor, *Segurança Informática no Software*, December 2016.

[20] Marc-André Cournoyer. Thin - A fast and very simple Ruby web server. http://code.macournoyer.com/thin/, March 2017.

[21] CVE Details - The MITRE Corporation. Vulnerabilities by year. https://www.cvedetails.com/browse-by-date.php, December 2017.

[22] CWE - The MITRE Corporation. A8. what is the relationship between cwe and cve? http://cwe.mitre.org/about/faq.html#A.8, December 2017.

[23] Nadeem Douba. Canari framework. http://www.canariproject.com/en/latest/, December 2017.

[24] Elastic. About Elastic. https://www.elastic.co, January 2018.

[25] Elastic. ElasticSearch: RESTful, Distributed Search & Analytics. https://www.elastic.co/products/elasticsearch, January 2018.

[26] Elastic. Kibana: Explore, Visualize, Discover Data. https://www.elastic.co/products/kibana, January 2018.

[27] Elasticsearch. Glossary of terms. https://www.elastic.co/guide/en/elasticsearch/reference/current/glossary.html, December 2017.

[28] PCMag. Encyclopedia. Ziff Davis. Cybersecurity. http://www.pcmag.com/encyclopedia/term/40169/computer-security, December 2016.

[29] FireEye. What is a zero-day exploit? https://www.fireeye.com/current-threats/what-is-a-zero-day-exploit.html, December 2017.

[30] FIRST.org, Inc. Complete cvss v1 guide. https://www.first.org/cvss/v1/guide, December 2017.

[31] Python Software Foundation. Python. https://www.python.org/about/, December 2017.

[32] Greenbone Networks. About OpenVAS Software. http://www.openvas.org/software.html#architecture_overview, January 2018.

[33] Greenbone Networks. OpenVAS - Open Vulnerability Assessment System. http://www.openvas.org/index.html, January 2018.

[34] IETF. The syslog protocol. https://tools.ietf.org/html/rfc5424, April 2017.

[35] InfoSec Institute. What is a SIEM? http://resources.infosecinstitute.com/what-is-a-siem/, January 2018.

[36] Zero-Day Initiative. Public disclosed vulnerabilities. http://www.zerodayinitiative.com/advisories/published/, December 2016.

[37] ISO/IEC. Iso/iec 27005:2011. In *Information technology — Security techniques — Information security risk management*, December 2016.

[38] Kaspersky. What is zero day exploit? https://www.kaspersky.com/resource-center/definitions/zero-day-exploit, December 2017.

[39] Kaspersky Lab. What is a botnet? - definition. https://www.kaspersky.com/resource-center/threats/botnet-attacks, December 2017.

[40] Mann and Christey. Towards a common enumeration of vulnerabilities. http://cve.mitre.org/docs/docs-2000/cerias.html, December 2017.

[41] WhatIs.com Margaret Rouse. Cybersecurity. http://whatis.techtarget.com/definition/cybersecurity, December 2016.

[42] WhatIs.com Margaret Rouse. Exploits. http://searchsecurity.techtarget.com/definition/exploit, December 2016.

[43] WhatIs.com Margaret Rouse. Malware. http://searchsecurity.techtarget.com/definition/malware, December 2016.

[44] WhatIs.com Margaret Rouse. confidentiality, integrity, and availability (CIA triad). https://whatis.techtarget.com/definition/Confidentiality-integrity-and-availability-CIA, December 2017.

[45] WhatIs.com Margaret Rouse. Patch. http://searchenterprisedesktop.techtarget.com/definition/patch, December 2017.

[46] Yukihiro Matsumoto. Ruby. https://www.ruby-lang.org/pt/, November 2016.

[47] Trend Micro. Syslog message formats. https://help.deepsecurity.trendmicro.com/Events-Alerts/syslog-parsing.html, April 2017.

[48] Trend Micro. 2017's notable vulnerabilities and exploits. https://www.trendmicro.com/vinfo/us/security/news/vulnerabilities-and-exploits/2017-notable-vulnerabilities-and-exploits, January 2018.

[49] Micro Focus. Enterprise Security Information and Event Management Solutions. https://software.microfocus.com/en-us/products/siem-security-information-event-management/overview, January 2018.

[50] NIST. About NIST. https://www.nist.gov/about-nist, December 2017.

[51] NIST - National Institute of Standards and Technology. Technical Guide to Information Security Testing and Assessment. http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf, January 2018.

[52] OWASP. Testing guide introduction. https://www.owasp.org/index.php/Testing_Guide_Introduction, December 2017.

[53] Paterva. Maltego. https://www.paterva.com/web7/index.php, December 2016.

[54] Paterva. What is maltego? https://www.paterva.com/web7/buy/maltego-clients/maltego.php, December 2016.

[55] Pivotal. RabbitMQ - Messaging that just works. https://www.rabbitmq.com, January 2018.

[56] PT. Potugal telecom. https://www.telecom.pt/pt-pt, November 2016.

[57] Inc. Qualys. Qualys cloud based. https://www.qualys.com/, November 2016.

[58] Rapid7. Rapid7 Vulnerability Scanner Tools. https://www.rapid7.com/products/insightvm/download/editions/, January 2018.

[59] Rapid7. Top Rated Vulnerability Management Software. https://www.rapid7.com/products/nexpose/, January 2018.

[60] Symantec. A new zero-day vulnerability discovered each week. https://www.symantec.com/security-center/threat-report, December 2016.

[61] Techopedia. Bug fix. https://www.techopedia.com/definition/18105/bug-fix, December 2017.

[62] Techopedia. Patch. https://www.techopedia.com/definition/24537/patch, December 2017.

[63] The MITRE Corporation. Corporate overview. https://www.mitre.org, December 2017.

[64] the free encyclopedia Wikipedia. Doublepulsar. https://en.wikipedia.org/wiki/DoublePulsar, December 2017.

[65] the free encyclopedia Wikipedia. Eternalblue. https://en.wikipedia.org/wiki/EternalBlue, December 2017.

[66] the free encyclopedia Wikipedia. Wannacry ransomware attack. https://en.wikipedia.org/wiki/WannaCry_ransomware_attack, December 2017.